Journal of the Royal Statistical Society
SERIES B Statistical Methodology

J. R. Statist. Soc. B (2011)
73, *Part* 1, *pp.* 3–36

Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models

Simon N. Wood

University of Bath, Bath, UK

[Received May 2009. Final revision April 2010]

Summary. Recent work by Reiss and Ogden provides a theoretical basis for sometimes preferring restricted maximum likelihood (REML) to generalized cross-validation (GCV) for smoothing parameter selection in semiparametric regression. However, existing REML or marginal likelihood (ML) based methods for semiparametric generalized linear models (GLMs) use iterative REML or ML estimation of the smoothing parameters of working linear approximations to the GLM. Such indirect schemes need not converge and fail to do so in a non-negligible proportion of practical analyses. By contrast, very reliable prediction error criteria smoothing parameter selection methods are available, based on direct optimization of GCV, or related criteria, for the GLM itself. Since such methods directly optimize properly defined functions of the smoothing parameters, they have much more reliable convergence properties. The paper develops the first such method for REML or ML estimation of smoothing parameters. A Laplace approximation is used to obtain an approximate REML or ML for any GLM, which is suitable for efficient direct optimization. This REML or ML criterion requires that Newton–Raphson iteration, rather than Fisher scoring, be used for GLM fitting, and a computationally stable approach to this is proposed. The REML or ML criterion itself is optimized by a Newton method, with the derivatives required obtained by a mixture of implicit differentiation and direct methods. The method will cope with numerical rank deficiency in the fitted model and in fact provides a slight improvement in numerical robustness on the earlier method of Wood for prediction error criteria based smoothness selection. Simulation results suggest that the new REML and ML methods offer some improvement in mean-square error performance relative to GCV or Akaike's information criterion in most cases, without the small number of severe undersmoothing failures to which Akaike's information criterion and GCV are prone. This is achieved at the same computational cost as GCV or Akaike's information criterion. The new approach also eliminates the convergence failures of previous REML- or ML-based approaches for penalized GLMs and usually has lower computational cost than these alternatives. Example applications are presented in adaptive smoothing, scalar on function regression and generalized additive model selection.

Keywords: Adaptive smoothing; Generalized additive mixed model; Generalized additive model; Generalized cross-validation; Marginal likelihood; Model selection; Penalized generalized linear model; Penalized regression splines; Restricted maximum likelihood; Scalar on function regression; Stable computation

1. Introduction

This paper is about reliable and efficient computation of likelihood-based smoothing parameter estimates in penalized generalized linear models (GLMs). Consider a GLM in which n independent univariate response variables y_i, with mean μ_i, depend on predictors via the model

Address for correspondence: Simon N. Wood, Department of Mathematical Sciences, University of Bath, Claverton Down, Bath, BA2 7AY, UK.
E-mail: s.wood@bath.ac.uk

 1369–7412/11/73003

$$g(\mu_i) = \mathbf{X}_i^* \boldsymbol{\beta}^* + \sum_j L_{ij} f_j, \qquad y_i \sim \text{an exponential family distribution}, \tag{1}$$

where g is a known monotonic link function, the f_j are smooth but unknown functions of any number of covariates, the L_{ij} are known linear functionals (usually dependent on covariates) and $\mathbf{X}_i^*$ is the ith row of the model matrix for any strictly parametric model components, with corresponding coefficients $\boldsymbol{\beta}^*$. Restriction to the exponential family implies that $\text{var}(y_i) = \phi V(\mu_i)$, for some known 'variance function' V and known or unknown 'scale parameter' ϕ. Typical $L_{ij} f_j$-terms are $f_j(x_i)$, $f_j(x_i) z_i$ or $\int f_j(x) k_i(x)\,\mathrm{d}x$ (where k_i is known), corresponding to generalized additive, varying coefficient and signal regression models respectively. For more on such models see, for example, Hastie and Tibshirani (1986, 1990), Ruppert *et al.* (2003), Wood (2006), Hastie and Tibshirani (1993), Marx and Eilers (1999), Ramsay and Silverman (2005), Reiss and Ogden (2007), Wahba (1990), Eilers and Marx (2002) and Fahrmeir *et al.* (2004).

To estimate model (1) in practice, the f_j can be represented by intermediate rank spline-type basis expansions (as originally proposed by Wahba (1980) and Parker and Rice (1985), for example), in which case the model becomes the GLM (Nelder and Wedderburn, 1972)

$$g(\mu_i) = \mathbf{X}_i \boldsymbol{\beta}, \qquad y_i \sim \text{ an exponential family distribution}, \tag{2}$$

where $\boldsymbol{\beta}$ now includes $\boldsymbol{\beta}^*$ and all the basis coefficients, and $\mathbf{X}$ is the corresponding $n \times q$ model matrix, with q usually substantially less than n. If the spline bases dimensions are sufficiently large to ensure reasonably low bias, then maximum likelihood estimation of model (2) will almost certainly lead to overfitting. To avoid this, the model should be estimated by penalized likelihood maximization, where the penalties suppress overly wiggly components f_j. In particular, the model is estimated by minimizing

$$D(\boldsymbol{\beta}) + \sum_j \lambda_j \boldsymbol{\beta}^{\mathrm{T}} \mathbf{S}_j \boldsymbol{\beta} \tag{3}$$

with respect to $\boldsymbol{\beta}$, where D is the model deviance, defined as the saturated log-likelihood minus the log-likelihood, all multiplied by 2ϕ (D is a useful GLM analogue of the residual sum of squares of a linear model, and working in terms of D will allow the direct use of some results from Wood (2008)); the $\mathbf{S}_j$ are $q \times q$ positive semidefinite matrices and the λ_j are positive smoothing parameters. Usually the $\boldsymbol{\beta}^{\mathrm{T}} \mathbf{S}_j \boldsymbol{\beta}$ measure the wiggliness of the f_j. In fact there may be several such penalties per f_j, e.g. when using tensor product (e.g. Wood (2006)) or adaptive (e.g. Krivobokova *et al.* (2008)) smoothing bases. The $\mathbf{S}_j$ may also be components of more general random-effects precision matrices.

Given the λ_j, there is a unique minimizer of expression (3), $\hat{\boldsymbol{\beta}}_\lambda$, which is straightforward to compute by a penalized version of the iteratively reweighted least squares method that is used for GLM estimation (penalized iteratively reweighted least squares (PIRLS)) (see for example Wood (2006) or Section 3.2). To select values for the λ_j requires optimization of a separate criterion, $\mathcal{V}(\boldsymbol{\lambda})$, say, which must be chosen.

1.1. Smoothness selection: prediction error or likelihood?

The λ_i selection criteria that have been proposed fall into two main classes. The first group try to minimize model prediction error, by optimizing criteria such as Akaike's information criterion (AIC), cross-validation or generalized cross-validation (GCV) (see for example Wahba and Wold (1975) and Craven and Wahba (1979)). The second group treat the smooth functions as random effects (Kimeldorf and Wahba, 1970), so that the λ_i are variance parameters which can be estimated by maximum (marginal) likelihood (ML) (Anderssen and Bloomfield, 1974), or restricted maximum likelihood (REML), which Wahba (1985) called 'generalized maximum likelihood'.

Asymptotically prediction error methods give better prediction error performance than likelihood-based methods (e.g. Wahba (1985) and Kauermann (2005)) but also have slower convergence of smoothing parameters to their optimal values (Härdle *et al.*, 1988). Reflecting this, published simulation studies (e.g. Wahba (1985), Gu (2002), Ruppert *et al.* (2003) and Kohn *et al.* (1991)) differ about the relative performance of the two classes, although there is agreement that prediction error criteria are prone to occasional severe undersmoothing. Reiss and Ogden (2009) provided a theoretical comparison of REML and GCV at *finite* sample sizes, showing that GCV is both more likely to develop multiple minima and to give more variable λ_j-estimates. Fig. 1 illustrates the basic issue. GCV penalizes overfit only weakly, with a minimum that tends to be very shallow on the undersmoothing side, relative to sampling variability. This can lead to an overfit. By contrast, REML (and also ML) penalizes overfit more severely and therefore tends to have a much more pronounced optimum, relative to sampling variability. In principle, extreme undersmoothing can also be avoided by use of modified prediction error criteria such as AICc (Hurvich *et al.*, 1998), but in practice the use of low to intermediate rank bases for the f_j already suppresses severe overfit, and AICc then offers little *additional* benefit relative to GCV, as Fig. 1 also illustrates.

Greater resistance to overfit, less smoothing parameter variability and a reduced tendency to multiple minima suggest that REML or ML might be preferable to GCV for semiparametric GLM estimation. But these benefits must be weighed against the fact that existing computational methods for REML or ML estimation of semiparametric GLMs are substantially less reliable than their prediction error equivalents, as the remainder of this section explains.

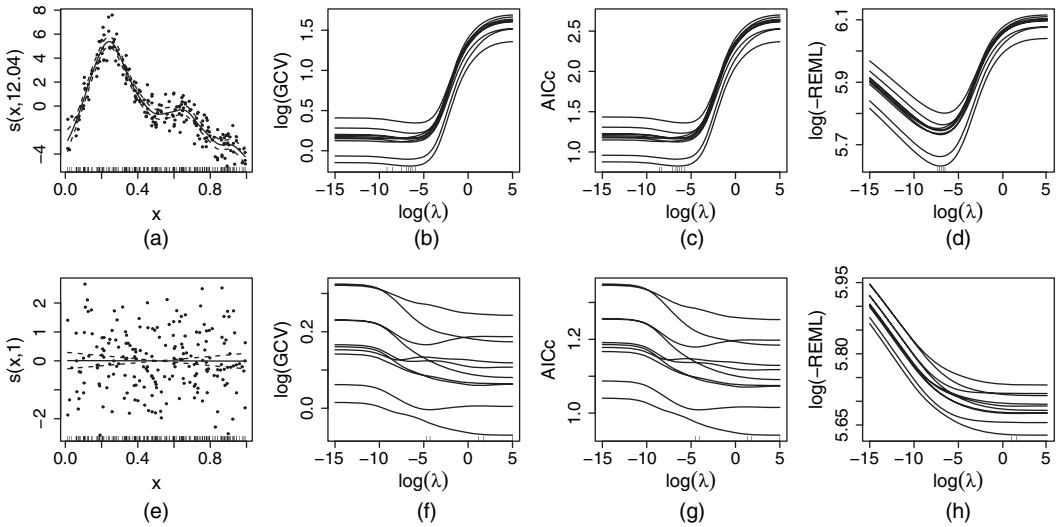

Fig. 1. Example comparison of GCV, AICc and REML criteria: (a) some (x,y)-data modelled as $y_i = f(x_i) + \varepsilon_i$, ε_i independent and identically distributed $N(0, \sigma^2)$ where smooth function f was represented by using a rank 20 thin plate regression spline (Wood, 2003); (b)–(d) various smoothness selection criteria plotted against logarithmic smoothing parameters, for 10 replicates of the data (each generated from the same 'truth') (note how shallow the GCV and AICc minima are relative to the sampling variability, resulting in rather variable optimal λ-values (which are shown as a rug plot), and a propensity to undersmooth; in contrast the REML optima are much better defined, relative to the sampling variability, resulting in a smaller range of λ-estimates); (e)–(h) are equivalent to (a)–(d), but for data with no signal, so that the appropriate smoothing parameter should tend to ∞ (note GCV's and AICc's occasional multiple minima and undersmoothing in this case, compared with the excellent behaviour of REML; ML (which is not shown) has a similar shape to REML)

There are two main classes of computational method for λ_j-estimation: those based on single iterations and those based on nested iterations. In the single-iteration case, each PIRLS step, which is used to update $\hat{\boldsymbol{\beta}}$, is supplemented by a $\hat{\boldsymbol{\lambda}}$-update. The latter is based on improving a $\boldsymbol{\lambda}$ selection criterion $\mathcal{V}_{\hat{\boldsymbol{\beta}}}(\boldsymbol{\lambda})$, which depends on the estimate of $\hat{\boldsymbol{\beta}}$ at the start of the step. $\mathcal{V}_{\hat{\boldsymbol{\beta}}}(\boldsymbol{\lambda})$ will be some sort of REML, GCV or similar criterion, but it is not a fixed function of $\boldsymbol{\lambda}$, instead changing with $\hat{\boldsymbol{\beta}}$ from iterate to iterate. Consequently single-iteration methods do not guarantee convergence to a fixed $\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\beta}}_{\hat{\boldsymbol{\lambda}}}$ (see Gu (2002), page 154, Wood (2006), page 180, and Brezger *et al.* (2007), reference manual section 8.1.2).

In nested iteration, the smoothness selection criterion $\mathcal{V}(\boldsymbol{\lambda})$ depends on $\boldsymbol{\beta}$ only via $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$. An outer iteration updates $\hat{\boldsymbol{\lambda}}$ to optimize $\mathcal{V}(\boldsymbol{\lambda})$, with each iterative step requiring an inner PIRLS iteration to find the current $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$. Because nested iteration optimizes a properly defined function of $\boldsymbol{\lambda}$, it is possible to guarantee convergence to a fixed optimum, provided that $\mathcal{V}$ is bounded below, and expression (3) has a well-defined optimum (conditions which are rather mild, in practice). The disadvantage of nested iteration is substantially increased computational complexity.

To date only single-iteration methods have been proposed for REML or ML estimation of semiparametric GLMs (e.g. Wood (2004), using Breslow and Clayton (1993), or Fahrmeir *et al.* (2004), using Harville (1977)), and in practice convergence problems are not unusual: examples are provided in Wood (2004, 2008), and in Appendix A. Early prediction-error-based methods were also based on single iteration (e.g. Gu (1992) and Wood (2004)), and suffered similar convergence problems, but these were overcome by Wood's (2008) nested iteration method for GCV, generalized approximate cross-validation, and AIC smoothness selection. Wood (2008) cannot be extended to REML or ML while maintaining good numerical stability, so the purpose of this paper is to provide an efficient and stable nested iteration method for REML or ML smoothness selection, thereby removing the major practical obstacle to use of these criteria.

2. Approximate restricted maximum likelihood or marginal likelihood for generalized linear model smoothing parameter estimation

Since the work of Kimeldorf and Wahba (1970), Wahba (1983) and Silverman (1985), it has been recognized that the penalized likelihood estimates $\hat{\boldsymbol{\beta}}$ are also the posterior modes of the distribution of $\boldsymbol{\beta}|\mathbf{y}$, if $\boldsymbol{\beta} \sim N(\mathbf{0}, \mathbf{S}^{-}\phi)$, where $\mathbf{S} = \Sigma_i \lambda_i \mathbf{S}_i$, and $\mathbf{S}^{-}$ is an appropriate generalized inverse (see for example Wood (2006)). Once the elements of $\boldsymbol{\beta}$ are viewed as random effects in this way, it is natural to try to estimate the λ_i, and possibly ϕ, by ML or REML (Wahba, 1985).

This preliminary section uses standard methods to obtain an approximate REML expression that is suitable for efficient direct optimization to estimate the smoothing parameters of a semiparametric GLM. Rather than follow Patterson and Thompson (1971) directly, Laird and Ware's (1982) approach to REML is taken, in which fixed effects are viewed as random effects with improper uniform priors and are integrated out. The key feature of the resulting expression is that it is relatively efficient to compute with and is suitable for optimizing as a properly defined function of the smoothing parameters, i.e., in contrast with previous single-iteration approaches to this problem, there is no need to resort to optimizing the REML score of a working model. Since a very similar approach obtains an approximate ML, this is also derived. ML can be useful for comparing models with different smooth terms included, for example (REML cannot be used for such a comparison because the alternative models will differ in fixed effect structure).

Consider a penalized GLM with log-likelihood $l(\boldsymbol{\beta}) = \log\{f_y(y|\boldsymbol{\beta})\}$. Under the random-effects formulation we have an improper 'prior' density for $\boldsymbol{\beta}$,

$$f_{\boldsymbol\beta}(\boldsymbol\beta) = \frac{|\mathbf{S}/\phi|_+^{0.5}}{\sqrt{(2\pi)^{n_b - M_p}}} \exp\left(\frac{-\boldsymbol\beta^{\mathrm{T}}\mathbf{S}\boldsymbol\beta}{2\phi}\right),$$

where $|\mathbf{B}|_+$ denotes the product of the non-zero eigenvalues of $\mathbf{B}$. n_b is the dimension of $\boldsymbol\beta$ and M_p is the dimension of the null space of $\mathbf{S}$. To obtain the restricted likelihood for REML we need to integrate $\boldsymbol\beta$ out of $f(y,\boldsymbol\beta) = f_y(y|\boldsymbol\beta)\, f_{\boldsymbol\beta}(\boldsymbol\beta)$ (for ML we would need to integrate out the part of $\boldsymbol\beta$ that is in the range space of $\mathbf{S}$). In practice the integral can be approximated as follows. Let $\mathbf{H} = -\partial^2 l/\partial\boldsymbol\beta\,\partial\boldsymbol\beta^{\mathrm{T}}$, and $\hat{\boldsymbol\beta}$ be the maximizer of $f(y,\boldsymbol\beta)$, i.e. the penalized likelihood estimates. Then

$$\begin{aligned} f(y,\boldsymbol\beta) &\simeq \exp[\log\{f_y(y|\hat{\boldsymbol\beta})\} + \log\{f_{\boldsymbol\beta}(\hat{\boldsymbol\beta})\} - (\boldsymbol\beta-\hat{\boldsymbol\beta})^{\mathrm{T}}(\mathbf{H}+\mathbf{S}/\phi)(\boldsymbol\beta-\hat{\boldsymbol\beta})/2] \\ &= f_y(y|\hat{\boldsymbol\beta})\, f_{\boldsymbol\beta}(\hat{\boldsymbol\beta}) \exp\{-(\boldsymbol\beta-\hat{\boldsymbol\beta})^{\mathrm{T}}(\mathbf{H}+\mathbf{S}/\phi)(\boldsymbol\beta-\hat{\boldsymbol\beta})/2\}. \end{aligned}$$

Integrating with respect to $\boldsymbol\beta$, and denoting the likelihood by L, we obtain the Laplace approximate REML criterion

$$L_{\mathrm{R}}(\lambda,\phi) = L(\hat{\boldsymbol\beta})\, f_{\boldsymbol\beta}(\hat{\boldsymbol\beta}) \frac{\sqrt{(2\pi)^{n_b}}}{|\mathbf{H}+\mathbf{S}/\phi|^{0.5}}$$

(which is actually exact for Gaussian models with the identity link), i.e., defining $l_r = \log(L_r)$,

$$2l_r = 2l(\hat{\boldsymbol\beta}) + \log(|\mathbf{S}/\phi|_+) - \hat{\boldsymbol\beta}^{\mathrm{T}}\mathbf{S}\hat{\boldsymbol\beta}/\phi - \log|\mathbf{H}+\mathbf{S}/\phi| + M_p\log(2\pi).$$

If the penalized GLM has its coefficients estimated by Newton-based PIRLS, as suggested below, then $\mathbf{H} = \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}/\phi$, where $\mathbf{W}$ is a diagonal weight matrix. To obtain ML, rather than REML, we would need to reparameterize to separate the parameters into penalized and unpenalized. Then $\mathbf{H}$ would be the negative Hessian for the penalized parameters only: further details are provided below in Section 2.1.

For ease of computation it helps to separate out l_r into ϕ-dependent and ϕ-independent components. For this, let $l_{\mathrm{s}}(\phi)$ denote the saturated log-likelihood and define

$$D_p = D(\hat{\boldsymbol\beta}) + \hat{\boldsymbol\beta}^{\mathrm{T}}\mathbf{S}\hat{\boldsymbol\beta}$$

and (assuming Newton weights)

$$K = \{\log|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S}| - \log(|\mathbf{S}|_+)\}/2.$$

We then have that

$$-l_r = \frac{D_p}{2\phi} - l_s(\phi) + K - \frac{M_p}{2}\log(2\pi\phi). \tag{4}$$

There are two approaches to the estimation of ϕ:

(a) estimate ϕ as part of l_r-maximization, or
(b) use the Pearson statistic over $n - M_p$ as $\hat{\phi}$, and optimize the resulting criterion, taking account of the derivatives of $\hat{\phi}$ with respect to the smoothing parameters.

The only advantage of approach (b) is that it may sometimes allow the resulting REML score to be used as a heuristic method of smoothness selection with quasi-likelihood.

The simpler approach of using the expected Hessian in place of $\mathbf{H}$ was also investigated, but in simulations it gave worse performance than GCV when non-canonical links were used.

2.1. Marginal likelihood details

For Laplace approximate ML, rather than REML, estimation, the only difference to the criterion is that we now need $\mathbf{H}$ to be the negative Hessian with respect to the coefficients of any orthogonal

basis for the range space of the penalty. The easiest way to separate out the range space is to form the eigendecomposition

$$\sum_j \mathbf{S}_j / \|\mathbf{S}_j\|_{\mathrm{F}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}},$$

where the scaling of each $\mathbf{S}_j$ by its Frobenius norm maintains good numerical conditioning. The first $q - M_p$ columns of $\mathbf{U}$ now form an orthogonal basis for the range space of $\mathbf{S}$ (see for example Wood (2006), sections 4.8.2 and 6.6.1). In consequence, if we reparameterize by setting $\bar{\boldsymbol{\beta}} = \mathbf{U}^{\mathrm{T}}\boldsymbol{\beta}$ then the first $q - M_p$ elements of $\bar{\boldsymbol{\beta}}$ will be penalized and should be integrated out of the joint density of $\mathbf{y}$ and $\bar{\boldsymbol{\beta}}$, whereas the last M_p elements are unpenalized, and hence left alone. Let $\mathbf{U}_1$ be the first $q - M_p$ columns of $\mathbf{U}$. Applying the reparameterization we have $\bar{\mathbf{X}} = \mathbf{X}\mathbf{U}_1$ and $\bar{\mathbf{S}} = \mathbf{U}_1^{\mathrm{T}}\mathbf{S}\mathbf{U}_1$, and some work establishes that the negative (Laplace approximate) log-marginal-likelihood is

$$-l = \frac{D_p}{2\phi} - l_s(\phi) + \frac{\log|\bar{\mathbf{X}}^{\mathrm{T}}\mathbf{W}\bar{\mathbf{X}} + \bar{\mathbf{S}}| - \log(|\mathbf{S}|_+)}{2}. \tag{5}$$

2.2. Accuracy of the Laplace approximation

For fixed dimension of $\boldsymbol{\beta}$, the true REML or ML integral divided by its Laplace approximation is $1 + O(n^{-1})$ (see for example Davison (2003), section 11.3.1). For consistency, it is usually necessary for the dimension of $\boldsymbol{\beta}$ to grow with n, which reduces this rate somewhat. However, for spline-type smoothers the dimension need only grow slowly with n (for example Gu and Kim (2002) showed that the rate need only be $O(n^{2/9})$ for cubic-spline-like smooths), so convergence is still rapid. Kauermann *et al.* (2009) showed in detail that the Laplace approximation is well justified asymptotically for ML in the penalized regression spline setting.

Rapid convergence does not in itself guarantee that the approximation is sufficiently accurate for any particular finite sample. Fortunately a simple and computationally efficient accuracy check is readily implemented, since a rather precise unbiased estimator of the REML score can be obtained by importance sampling with a 'Laplace proposal'. In particular, if $\mathbf{R}$ is a square factor such that

$$\mathbf{R}^{\mathrm{T}}\mathbf{R} = (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S})^{-1}\hat{\phi},$$

and $\mathbf{z}_i$ are n_s independent $N(\mathbf{0}, \mathbf{I})$ random n_b vectors, then

$$\frac{(2\pi)^{n_b/2}}{n_s|\mathbf{R}|} \sum_{i=1}^{n_s} f_y(y|\hat{\boldsymbol{\beta}} + \mathbf{R}^{\mathrm{T}}\mathbf{z}_i)\, f_{\boldsymbol{\beta}}(\hat{\boldsymbol{\beta}} + \mathbf{R}^{\mathrm{T}}\mathbf{z}_i) \exp\left(\frac{\|\mathbf{z}_i\|^2}{2}\right)$$

is an unbiased estimator of the exact REML score (see, for example, Monahan (2001), section 10.4C). In the work that is reported here n_s in the range 1000–10000 was sufficient to ensure that the Monte Carlo variability was at least an order of magnitude smaller than the mean difference between the estimator and the deterministic Laplace approximation. This estimator was used to estimate the Laplace approximation error, at the estimated smoothing parameters, for all the examples that are presented subsequently in this paper. The worst error was for the binary simulations in Section 4, where the magnitude of the error was up to 0.3. The other examples had approximation errors that were an order of magnitude smaller. Hence the error that is induced by the deterministic Laplace approximation is not significant relative to the sampling uncertainty in the smoothing parameters, suggesting that the Laplace approximation is adequate for the examples that are presented here.

Note that the Laplace approximation that is employed here does not suffer from the difficulties that are common to most penalized quasi-likelihood (PQL) (Breslow and Clayton, 1993)

implementations when used with binary data. Most PQL implementations must estimate ϕ for the working model, even with binary data where this is not really satisfactory. In addition, PQL uses the expected Hessian in place of the exact Hessian when non-canonical links are used, which also reduces accuracy. That said, it should still be expected that the accuracy of equations (4) and (5) will reduce for binary or Poisson data when the expectation of the response variable is very low.

3. Optimizing the restricted maximum likelihood criterion

Equations (4) and (5) depend on the smoothing parameter vector $\boldsymbol{\lambda}$ via the dependence of $\mathbf{S}$ and $\hat{\boldsymbol{\beta}}$ (and hence $\mathbf{W}$) on $\boldsymbol{\lambda}$. The proposal here is to optimize equation (4) or (5) with respect to the $\rho_i = \log(\lambda_i)$, by using Newton's method, with the usual modifications that

(a) some step length control will be used and
(b) the Hessian will be perturbed to be positive definite, if it is not (see Nocedal and Wright (2006) for an up-to-date treatment and computational details).

Each trial logarithmic smoothing parameter vector $\boldsymbol{\rho}$, proposed as part of the Newton method iteration, will require a PIRLS iteration to evaluate the corresponding $\hat{\boldsymbol{\beta}}$ (and hence $\mathbf{W}$). So the whole optimization consists of two nested iterations: an outer iteration to find $\hat{\boldsymbol{\rho}}$, and an inner iteration to find the $\hat{\boldsymbol{\beta}}$ corresponding to any $\boldsymbol{\rho}$. The outer iteration requires the gradient and Hessian of equation (4) or (5) with respect to $\boldsymbol{\rho}$, and this in turn requires first and second derivatives of $\hat{\boldsymbol{\beta}}$ with respect to $\boldsymbol{\rho}$.

Irrespective of the details of the optimization method, the major difficulty in minimizing equation (4) or (5) is that, if some λ_j is sufficiently large, then the 'numerical footprint' of the corresponding penalty term $\lambda_j \boldsymbol{\beta}^{\mathrm{T}} \mathbf{S}_j \boldsymbol{\beta}$ can extend well beyond the penalty's range space, i.e. numerically the penalty can have marked effects in the subspace of the model parameter space for which, formally, $\boldsymbol{\beta}^{\mathrm{T}} \mathbf{S}_j \boldsymbol{\beta} = 0$. For example if $\|\lambda_j \mathbf{S}_j\| \gg \|\lambda_k \mathbf{S}_k\|$ then $\lambda_j \mathbf{S}_j$ can have effects which are 'numerically zero' when judged relatively to $\|\lambda_j \mathbf{S}_j\|$ (and would be exactly zero in infinite precision arithmetic), but which are larger than the strictly non-zero effects of $\lambda_k \mathbf{S}_k$. If left uncorrected, this problem leads to serious errors in evaluation of $\hat{\boldsymbol{\beta}}$, $|\mathbf{S}|_+$ and $|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S}|$ and their derivatives with respect to $\boldsymbol{\rho}$ (see Section 3.1). Because multiple penalties often have overlapping range spaces (i.e. they penalize intersecting subspaces of the parameter space), no single reparameterization can solve this problem for all $\boldsymbol{\lambda}$-values, but an adaptive reparameterization approach does work and is outlined in Section 3.1. Note that the Wood (2008) method, for dealing with numerical ill conditioning for prediction error criteria, is hopeless here. That method truncates the parameter space to deal with ill conditioning that is induced by changes in $\boldsymbol{\lambda}$, but such an approach would lead to large erroneous and discontinuous changes in $|\mathbf{S}|_+$ and $|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S}|$ as $\boldsymbol{\lambda}$ changes. We shall of course still need to truncate the parameter space if some parameters would not be identifiable whatever the value of $\boldsymbol{\lambda}$, but such a $\boldsymbol{\lambda}$-independent truncation is not problematic.

A second question, when minimizing equation (4) or (5), is what optimization method to use to obtain the $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$ corresponding to any trial $\boldsymbol{\lambda}$. If a PIRLS scheme is employed based on Newton (rather than Fisher) updates, then the Hessian that is required in equation (4) or (5) is conveniently obtained as a by-product of fitting, which also means that the same method can be used to stabilize both $\hat{\boldsymbol{\beta}}$ and REML or ML evaluation. Furthermore the required derivatives of $\hat{\boldsymbol{\beta}}$ with respect to $\boldsymbol{\rho}$ can be obtained directly from the information that is available as part of the PIRLS, using implicit differentiation, without the need for further iteration. Newton-based PIRLS also leads to more rapid convergence with non-canonical links.

As a result of the preceding considerations, this paper proposes that the following steps should be taken for each trial $\boldsymbol{\rho}$ proposed by the outer Newton iteration.

Step 1: reparameterize to avoid large norm $\lambda_j\mathbf{S}_j$-terms having effects outside their range spaces, thereby ensuring accurate computation with the current $\boldsymbol{\rho}$ (Section 3.1).
Step 2: estimate $\hat{\boldsymbol{\beta}}$ by Newton-based PIRLS, setting to 0 any elements of $\hat{\boldsymbol{\beta}}$ which would be unidentifiable *irrespectively of the value of* $\boldsymbol{\rho}$ (Sections 3.2 and 3.3).
Step 3: obtain first and second derivatives of $\hat{\boldsymbol{\beta}}$ with respect to $\boldsymbol{\rho}$, using implicit differentiation and the quantities that are calculated as part of step 2 (Section 3.4).
Step 4: using the results from steps 2 and 3, evaluate the REML or ML criterion and derivatives with respect to $\boldsymbol{\rho}$ (Section 3.5).

After these four steps, all the ingredients are in place to propose a new $\boldsymbol{\rho}$ by using a further step of Newton's method.

3.1. Reparameterization, log $|\mathbf{S}|_+$ and $\sqrt{\mathbf{S}}$

$\log(|\mathbf{S}|_+)$ (where $\mathbf{S}=\Sigma_j \lambda_j\mathbf{S}_j$) is the most numerically troublesome term in the REML or ML objective. Both $\lambda_i \to 0$ *and* $\lambda_i \to \infty$ can cause numerical problems when evaluating the determinant. The problem is most easily seen by considering the simple example of evaluating $|\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2|$ when the $q \times q$ positive semidefinite dense matrices $\mathbf{S}_j$ are not full rank, but $\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2$ is. In what follows let $\|\cdot\|$ denote the matrix 2-norm (although the 1-, ∞- or Frobenius norms would serve as well), and let $\hat{x}$ denote the computed version of any quantity x. Consider a similarity transform based on the eigendecomposition $\mathbf{S}_1 = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}$, with computed version $\mathbf{S}_1 = \hat{\mathbf{U}}\hat{\boldsymbol{\Lambda}}\hat{\mathbf{U}}^{\mathrm{T}}$. Let $\boldsymbol{\Lambda}^+$ denote the vector of strictly positive eigenvalues, and $\boldsymbol{\Lambda}^0$ the vector of zero eigenvalues, and note that $\hat{\boldsymbol{\Lambda}}^0$ will have elements of typical magnitude $\|\mathbf{S}_1\|\varepsilon_m$ where ε_m is the computational machine precision (see for example Watkins (1991), section 5.5, or Golub and van Loan (1996), chapter 8).

By standard properties of similarity transforms we have

$$|\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2| = |\lambda_1\boldsymbol{\Lambda} + \lambda_2\mathbf{U}^{\mathrm{T}}\mathbf{S}_2\mathbf{U}|. \tag{6}$$

Suppose that $\mathbf{S}_j$ has rank r_j and rank deficiency $d_j = q - r_j$. As $\lambda_1/\lambda_2 \to \infty$ it is routine that the r_1 largest eigenvalues of $\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2 \to \lambda_1\boldsymbol{\Lambda}^+$, so

$$|\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2| \to \lambda_1^{r_1}\prod_i \Lambda_i^+\alpha,$$

where the factor α depends on $\lambda_2\mathbf{S}_2$. However, as $\lambda_1/\lambda_2 \to \infty$ *all* the *computed* eigenvalues of $\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2 \to \lambda_1\hat{\boldsymbol{\Lambda}}$, so

$$|\lambda_1\widehat{\mathbf{S}_1 + \lambda_2}\mathbf{S}_2| \to \lambda_1^{r_1}\prod_i \hat{\Lambda}_i^+\lambda_1^{d_1}\prod_i \hat{\Lambda}_i^0.$$

Hence the computed determinant is seriously in error because the factor $\lambda_1^{d_1}\Pi_i\,\hat{\Lambda}_i^0$ is essentially arbitrary and is unrelated to the correct factor α. (Note that the problem vanishes for a full rank $\mathbf{S}_1$.)

The difficulty arises because the computed version of the matrix $\lambda_1\boldsymbol{\Lambda} + \lambda_2\mathbf{U}^{\mathrm{T}}\mathbf{S}_2\mathbf{U}$ is perturbed by the completely arbitrary error terms in $\lambda_1\hat{\boldsymbol{\Lambda}}^0$. In general the effect of a perturbation on the determinant of a positive definite $\mathbf{A}$, with eigenvalues $\boldsymbol{\Lambda}^{\mathrm{A}}$, depends on the size of the perturbation relative to $\min(\boldsymbol{\Lambda}^{\mathrm{A}})$. This is easily seen by considering a simple additive perturbation $\varepsilon\mathbf{I}$ (where ε is the size of perturbation). Then

$$|\mathbf{A} + \varepsilon\mathbf{I}|/|\mathbf{A}| = \prod_i (\Lambda_i^{\mathrm{A}} + \varepsilon)/\Lambda_i^{\mathrm{A}},$$

where the largest contribution to the right-hand side is from the term $\{\min(\mathbf{\Lambda}^{\mathrm{A}})+\varepsilon\}/\min(\mathbf{\Lambda}^{\mathrm{A}})$. Hence we can expect problems when the perturbations $\lambda_1\hat{\mathbf{\Lambda}}^0$ become non-negligible relative to the smallest eigenvalue of $\lambda_1\mathbf{S}_1+\lambda_2\mathbf{S}_2$, which is bounded below by the smallest positive eigenvalue of $\lambda_2\mathbf{S}_2$ as $\lambda_1/\lambda_2\to\infty$.

In short, we can expect this 'numerical zero leakage' issue to spoil determinant calculations whenever the ratio of the largest strictly positive eigenvalue of $\lambda_1\mathbf{S}_1$ (which sets the scale of the arbitrary perturbation, $\lambda_1\hat{\mathbf{\Lambda}}^0$) to the smallest strictly positive eigenvalue of $\lambda_2\mathbf{S}_2$ is too great. However, the example also suggests a simple way of suppressing the problem. Reparameterize by using the computed eigenbasis of the dominant term $\mathbf{S}_1$, so that $\mathbf{S}_1$ becomes $\hat{\mathbf{\Lambda}}$ and $\mathbf{S}_2$ becomes $\hat{\mathbf{U}}^{\mathrm{T}}\mathbf{S}_2\hat{\mathbf{U}}$. In the transformed space it is easy to ensure that the dominant term (now $\hat{\mathbf{\Lambda}}$) acts only within its range space, by setting $\hat{\mathbf{\Lambda}}^0=\mathbf{0}$ (if the rank of $\mathbf{S}_1$ is known then identifying which eigenvalues should be 0 is trivial; if not, see step 3 in Appendix B).

Having reparameterized and truncated in this way, stable evaluation of $|\lambda_1\mathbf{\Lambda}+\lambda_2\mathbf{U}^{\mathrm{T}}\mathbf{S}_2\mathbf{U}|$ is straightforward. Only the first r_1 columns of $\lambda_1\hat{\mathbf{\Lambda}}+\lambda_2\hat{\mathbf{U}}^{\mathrm{T}}\mathbf{S}_2\hat{\mathbf{U}}$ now depend on $\lambda_1\mathbf{S}_1$. Forming a pivoted QR-decomposition $\lambda_1\hat{\mathbf{\Lambda}}+\lambda_2\hat{\mathbf{U}}^{\mathrm{T}}\mathbf{S}_2\hat{\mathbf{U}}=\hat{\mathbf{Q}}\hat{\mathbf{R}}$ maintains this column separation in $\hat{\mathbf{R}}$ (the decomposition acts on columns, without mixing between columns), with the result that $|\lambda_1\widehat{\mathbf{S}_1+\lambda_2\mathbf{S}_2}|=|\lambda_1\hat{\mathbf{\Lambda}}+\lambda_2\hat{\mathbf{U}}^{\mathrm{T}}\mathbf{S}_2\hat{\mathbf{U}}|=\Pi_i\hat{R}_{ii}$ can be accurately computed. Furthermore, pivoting ensures that $\hat{\mathbf{R}}^{-1}$ is computable, which is necessary for derivative calculations. See Golub and van Loan (1996) for full discussion of QR-decomposition with pivoting.

The stable computation of $\hat{\boldsymbol{\beta}}$, which was discussed in Section 3.3, will also require that a square root of $\mathbf{S}$ can be formed that maintains the required 'column separation' of the dominant terms in $\mathbf{S}$ (i.e. we must not end up with large magnitude elements in some column $j>r_1$, just because $\lambda_1\|\mathbf{S}_1\|$ is large). This is quite straightforward under the reparameterization that was just discussed. For example, let $\hat{\mathbf{S}}'=\lambda_1\hat{\mathbf{\Lambda}}+\lambda_2\hat{\mathbf{U}}^{\mathrm{T}}\mathbf{S}_2\hat{\mathbf{U}}$ (with $\hat{\mathbf{\Lambda}}$s 'machine zeros' set to true zeros) and $\hat{\mathbf{P}}$ be the diagonal matrix such that $\hat{P}_{ii}=\sqrt{|\hat{S}'_{ii}|}$. Forming the Choleski decomposition $\hat{\mathbf{L}}\hat{\mathbf{L}}^{\mathrm{T}}=\hat{\mathbf{P}}^{-1}\hat{\mathbf{S}}'\hat{\mathbf{P}}^{-1}$, then $\hat{\mathbf{E}}=\hat{\mathbf{L}}^{\mathrm{T}}\hat{\mathbf{P}}$ is a matrix square root such that $\hat{\mathbf{E}}^{\mathrm{T}}\mathbf{E}=\hat{\mathbf{S}}'$. Furthermore, $\lambda_1\mathbf{S}_1$ affects only the size of the elements in $\hat{\mathbf{E}}$'s first r_1 columns (this is easily seen, since, from the definition of $\hat{\mathbf{E}}$, the squared Euclidean norm of its jth column is given by $\hat{\mathbf{S}}'_{jj}$, which does not depend on $\lambda_1\mathbf{S}_1$ if $j>r_1$). The preconditioning (or 'scaling') matrix $\hat{\mathbf{P}}^{-1}$ ensures that the Choleski factor can be computed in finite precision, however divergent the sizes of the components of $\mathbf{S}$ (see for example Watkins (1991), section 2.9). From now on no further purpose is served by distinguishing between 'true' and computed quantities, so circumflexes will be omitted.

Of course $\mathbf{S}=\Sigma\lambda_i\mathbf{S}_i$ generally contains more than two terms and is not full rank, but Appendix B generalizes the similarity-transform-based reparameterization, along with the (generalized) determinant and square-root calculations, to any number of components of a rank deficient $\mathbf{S}$. It also provides the expressions for the derivatives of $\log(|\mathbf{S}|_+)$ with respect to $\boldsymbol{\rho}$. The operations count for Appendix B is $O(q^3)$.

The stable matrix square root $\mathbf{E}$, produced by the Appendix B method, is only useful if the rest of the model fitting adopts the Appendix B reparameterization, i.e. the transformed $\mathbf{S}_i$, $\mathbf{S}$ and $\mathbf{E}$, computed by Appendix B, must be used in place of the original untransformed versions, along with a transformed version of the model matrix. To compute the latter, let $\mathbf{Q}_s$ be the orthogonal matrix describing the similarity transform applied by Appendix B, i.e., if $\mathbf{S}$ is the transformed total penalty matrix, then formally $\mathbf{Q}_s\mathbf{S}\mathbf{Q}_s^{\mathrm{T}}$ is the untransformed original. Then the transformed model matrix should be $\mathbf{X}\mathbf{Q}_s$ (obtained at $O(nq^2)$ cost). In what follows it is assumed that this reparameterization is always adopted, being recomputed for each new $\boldsymbol{\rho}$-value. So the model matrix and penalty matrices are taken to be the transformed versions, from now on. If the coefficient estimates in this parameterization are $\hat{\boldsymbol{\beta}}$, then the estimates in the original parameterization are $\mathbf{Q}_s\hat{\boldsymbol{\beta}}$.

Finally, reparameterization is preferable to simply limiting the working $\boldsymbol{\lambda}$-range. To keep the non-zero eigenvalues of all $\lambda_i \mathbf{S}_i$ within limits that guarantee computational stability usually entails unacceptably restrictive limits on the λ_i, i.e. limits that are sufficiently restrictive to ensure numerical stability have statistically noticeable effects.

3.2. Estimating the regression coefficients given smoothing parameters

Minimizing expression (3) by Newton's method or Fisher scoring both result in a PIRLS method, as follows. Pseudodata and weights are defined first:

$$z_i = \eta_i + \frac{(y_i - \mu_i) g_i'}{\alpha_i},$$
$$w_i = \frac{\omega_i \alpha_i}{V_i g_i'^2},$$

where $\eta_i = g(\mu_i) = \mathbf{X}_i \boldsymbol{\beta}$, $V_i = V(\mu_i)$,

$$\alpha_i = \begin{cases} 1 + (y_i - \mu_i)(V_i'/V_i + g_i''/g_i') & \text{for Newton's method,} \\ 1 & \text{for Fisher scoring} \end{cases}$$

and x' denotes $\mathrm{d}x/\mathrm{d}\mu_i$, whatever x. These quantities are always evaluated at the current μ_i-estimates. The ω_i are any prior weights and are usually 1. If a canonical link function is used then $\alpha_i = 1, \forall i$, and Newton's method and Fisher scoring coincide.

Estimation of the coefficients $\boldsymbol{\beta}$ is performed by the modified IRLS scheme of iterating the following two steps to convergence ($\boldsymbol{\mu}$-estimates are initialized by using the previous $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$, or directly from $\mathbf{y}$).

Step 1: given the current *estimate* of $\boldsymbol{\mu}$ (and hence $\boldsymbol{\eta}$), evaluate $\mathbf{z}$ and $\mathbf{w}$.

Step 2: solve the weighted penalized least squares problem of minimizing

$$\sum_{i=1}^{n} w_i (z_i - \mathbf{X}_i \boldsymbol{\beta})^2 + \sum_j \lambda_j \boldsymbol{\beta}^{\mathrm{T}} \mathbf{S}_j \boldsymbol{\beta} \tag{7}$$

with respect to $\boldsymbol{\beta}$, to obtain the updated estimate of $\boldsymbol{\beta}$ and hence $\boldsymbol{\mu}$ (and $\boldsymbol{\eta}$). See Section 3.3.

At convergence of the Newton-type iteration the Hessian of the deviance with respect to $\boldsymbol{\beta}$ is given by $2\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}$, where $\mathbf{W} = \mathrm{diag}(w_i)$. Under Fisher scoring $2\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}$ is the *expected* Hessian. See for example Green and Silverman (1994) or Wood (2006) for further information on (Fisher-based) PIRLS.

Several points should be noted.

(a) Step halving will be needed in the event that the penalized deviance increases at any iteration, but the Newton method should never require it at the end of the iteration.
(b) The Newton scheme tends to converge faster than Fisher scoring in non-canonical link situations, an effect which can be particularly marked when using Tweedie (1984) distributions.
(c) With non-canonical links, the w_i need not all be positive for the Newton scheme, and in practice negative weights are encountered for perfectly reasonable models: the next section deals with this. Negative w_i provide the second reason that the Wood (2008) method cannot be extended to REML.

3.3. Stable least squares with negative weights

This section develops a method for stable computation of weighted least squares problems when some weights are negative, as required by the Newton-based PIRLS that was described

in Section 3.2. The method also deals with identifiability problems that do not depend on the magnitude of $\boldsymbol{\lambda}$.

The obvious approach to solving expression (7) in the presence of negative weights would be to solve directly

$$(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S})\hat{\boldsymbol{\beta}}=\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{z} \tag{8}$$

for $\hat{\boldsymbol{\beta}}$, where $\mathbf{W}=\mathrm{diag}(w_i)$, $\mathbf{z}$ is the vector of z_i from Section 3.2 and $\mathbf{S}=\Sigma_j\lambda_j\mathbf{S}_j$. However, it is well known that direct formation of $\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}$ results in a system with a condition number that is the square of what is necessary (see for example Golub and van Loan (1996), sections 5.3.2 and 5.3.8). Given that penalized GLMs are frequently complex models in which concurvity effects can easily lead to quite high condition numbers, this approach is not sensible.

When weights are non-negative, a stable solution of equation (8) is based on orthogonal decomposition of $\sqrt{\mathbf{W}}\mathbf{X}$ (e.g. Wood (2004)), but this does not work if some weights are negative. This section proposes a stable solution method, by starting with a 'nearby' penalized least squares problem, for which all the weights are non-negative, applying a stable orthogonal decomposition approach to this, but at the same time developing the correction terms that are necessary to end up with the solution to equation (8) itself.

To make progress then, let $\mathbf{W}^{-}$ denote the diagonal matrix such that W^{-}_{ii} equals 0 if $w_i\geqslant 0$ and $-w_i$ otherwise. Also let $\bar{\mathbf{W}}$ be a diagonal matrix with $\bar{W}_{ii}=|w_i|$. In this case

$$\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}=\mathbf{X}^{\mathrm{T}}\bar{\mathbf{W}}\mathbf{X}-2\mathbf{X}^{\mathrm{T}}\mathbf{W}^{-}\mathbf{X}.$$

So $\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}$ has been split into a component that is straightforward to compute with stably, and a 'correction' term. Starting with the straightforward term, perform a QR-decomposition

$$\sqrt{\bar{\mathbf{W}}}\mathbf{X}=\mathcal{Q}\mathcal{R} \tag{9}$$

(either without pivoting, or reversing the pivoting of $\mathcal{R}$ after the decomposition). At this stage it is necessary to test for any inherent lack of identifiability in the problem (i.e. lack of identifiability which is $\boldsymbol{\lambda}$ independent). Section 3.3.1 describes how to do this. For the moment suppose that the inherent rank of the problem is r, and we have a list of any unidentifiable parameters. Then drop the columns of $\mathcal{R}$ and $\mathbf{X}$ and the rows and columns of the $\mathbf{S}_i$ corresponding to any unidentifiable parameters.

$\mathcal{R}$ is now a square root of $\mathbf{X}^{\mathrm{T}}\bar{\mathbf{W}}\mathbf{X}$, but we really need a square root of $\mathbf{X}^{\mathrm{T}}\bar{\mathbf{W}}\mathbf{X}+\mathbf{S}$, to move towards solution of equation (8). For this, let $\mathbf{E}$ be a matrix such that $\mathbf{E}^{\mathrm{T}}\mathbf{E}=\mathbf{S}$, computed as described in Appendix B and Section 3.1. Drop the columns of $\mathbf{E}$ corresponding to any unidentifiable parameters, and form a further pivoted QR-decomposition

$$\begin{pmatrix}\mathcal{R}\\ \mathbf{E}\end{pmatrix}=\mathbf{Q}\mathbf{R}. \tag{10}$$

$\mathbf{R}$ is the required pivoted square root of $\mathbf{X}^{\mathrm{T}}\bar{\mathbf{W}}\mathbf{X}+\mathbf{S}$. Now define $n\times r$ matrix $\mathbf{Q}_1=\mathcal{Q}\mathbf{Q}[1:q,]$, where q is the number of columns of $\mathbf{X}$ and $\mathbf{Q}[1:q,]$ denotes the first q rows of $\mathbf{Q}$. Hence

$$\sqrt{\bar{\mathbf{W}}}\mathbf{X}=\mathbf{Q}_1\mathbf{R}. \tag{11}$$

For what follows, the pivoting that is used in the QR-step (10) will have to be applied to the rows and columns of $\mathbf{S}_j$ and the columns of $\mathbf{X}$.

Now we need to correct the matrix square root $\mathbf{R}$ to obtain what is actually needed to solve equation (8):

$$\begin{aligned}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S} &= \mathbf{R}^{\mathrm{T}}\mathbf{R}-2\mathbf{X}^{\mathrm{T}}\mathbf{W}^{-}\mathbf{X}\\ &= \mathbf{R}^{\mathrm{T}}(\mathbf{I}-2\mathbf{R}^{-\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{W}^{-}\mathbf{X}\mathbf{R}^{-1})\mathbf{R}\\ &= \mathbf{R}^{\mathrm{T}}(\mathbf{I}-2\mathbf{R}^{-\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}}\mathbf{I}^{-}\mathbf{Q}_1\mathbf{R}\mathbf{R}^{-1})\mathbf{R}\\ &= \mathbf{R}^{\mathrm{T}}(\mathbf{I}-2\mathbf{Q}_1^{\mathrm{T}}\mathbf{I}^{-}\mathbf{Q}_1)\mathbf{R},\end{aligned}$$

where $\mathbf{I}^{-}$ denotes the diagonal matrix such that I^{-}_{ii} equals 0 if $w_i > 0$ and 1 otherwise, and $\mathbf{W}^{-} = \mathbf{I}^{-}\bar{\mathbf{W}}$. The matrix $\mathbf{I}-2\mathbf{Q}_1^{\mathrm{T}}\mathbf{I}^{-}\mathbf{Q}_1$ is not necessarily positive semidefinite and so requires careful handling. Forming the singular value decomposition

$$\mathbf{I}^{-}\mathbf{Q}_1 = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}} \tag{12}$$

(of course, in practice the zero rows of $\mathbf{I}^{-}\mathbf{Q}_1$ can be dropped before decomposition) then we obtain

$$\begin{aligned}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S} &= \mathbf{R}^{\mathrm{T}}(\mathbf{I}-2\mathbf{V}\mathbf{D}^2\mathbf{V}^{\mathrm{T}})\mathbf{R}\\ &= \mathbf{R}^{\mathrm{T}}\mathbf{V}(\mathbf{I}-2\mathbf{D}^2)\mathbf{V}^{\mathrm{T}}\mathbf{R}\end{aligned} \tag{13}$$

(and additionally $\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} = \mathcal{R}^{\mathrm{T}}\mathcal{R} - 2\mathbf{R}^{\mathrm{T}}\mathbf{V}\mathbf{D}^2\mathbf{V}^{\mathrm{T}}\mathbf{R}$). Now define

$$\begin{aligned}\mathbf{P} &= \mathbf{R}^{-1}\mathbf{V}(\mathbf{I}-2\mathbf{D}^2)^{-1/2},\\ \mathbf{K} &= \mathbf{Q}_1\mathbf{V}(\mathbf{I}-2\mathbf{D}^2)^{-1/2}.\end{aligned} \tag{14}$$

If $\bar{\mathbf{z}}$ is the vector such that $\bar{z}_i = z_i$ if $w_i \geqslant 0$ and $\bar{z}_i = -z_i$ otherwise, then substituting from equations (14), (13) and (11) into (8) and solving gives

$$\hat{\boldsymbol{\beta}} = \mathbf{P}\mathbf{K}^{\mathrm{T}}\sqrt{\bar{\mathbf{W}}}\bar{\mathbf{z}}.$$

The key point about this calculation is that its condition number will be dominated by that of $\mathbf{R}$, the matrix which must be inverted in the definition of $\mathbf{P}$. This is approximately the square root of the condition number for using $\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S}$ directly, since the term to be inverted in this latter case would be dominated by $\mathbf{R}^{\mathrm{T}}\mathbf{R}$ (see Golub and van Loan (1996), sections 2.7.2 and 3.5.4 if this is unclear). The key computational steps that are involved in finding $\hat{\boldsymbol{\beta}}$ are equations (9), (10), (12) and (14), plus the rank identification of Section 3.3.1.

Given equation (13), it is now possible to compute one of the REML log-determinant components by using

$$|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S}| = |\mathbf{R}|^2|\mathbf{I}-2\mathbf{D}^2|,$$

and it is also worth noting, from equations (13) and (14), that $(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S})^{-1} = \mathbf{P}\mathbf{P}^{\mathrm{T}}$ (strictly some sort of pseudoinverse if there is rank deficiency).

There is an important additional detail. At the penalized MLE, $\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S}$ will be positive semidefinite, so $d_i \leqslant 1/\sqrt{2}$ (reparameterize so that $\mathbf{R}$ is the identity to see this), but *en route* to the optimum there is no *guarantee* that the penalized likelihood is positive semidefinite. So, if $d_i > 1/\sqrt{2}$, for any i, then a Fisher step should be substituted, i.e. set $\alpha_i = 1$, so that $w_i \geqslant 0, \forall i$. Then

$$\mathbf{P} = \mathbf{R}^{-1} \quad \text{and} \quad \mathbf{K} = \mathbf{Q}_1$$

and the expression for $\hat{\boldsymbol{\beta}}$, above, simplifies to $\hat{\boldsymbol{\beta}} = \mathbf{P}\mathbf{K}^{\mathrm{T}}\sqrt{\mathbf{W}}\mathbf{z}$, while $|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S}| = |\mathbf{R}|^2$.

At the end of model fitting, $\hat{\boldsymbol{\beta}}$ will need to have the pivoting that was applied at equation (10) reversed, and the elements of $\hat{\boldsymbol{\beta}}$ that were dropped by the truncation step after equation (9) will have to be reinserted as 0s. Note that the leading order cost of the method that is described here is the $O(nq^2)$ of the first QR-decomposition. LAPACK can be used for all decompositions (Anderson *et al.*, 1999).

3.3.1. λ-independent rank deficiency

As mentioned above, it is necessary to deal with any rank deficiency of the weighted penalized least squares problem that is 'structural' to the problem, rather than being the numerical consequence of some smoothing parameter tending to 0 or ∞, i.e. we need to find which, if any, parameters $\boldsymbol{\beta}$ would be unidentifiable, even if the penalties and models matrix were all evenly scaled relative to each other.

To achieve this, first find $\bar{\mathbf{E}}$, a matrix such that

$$\bar{\mathbf{E}}^{\mathrm{T}}\bar{\mathbf{E}} = \sum_i \mathbf{S}_i / \|\mathbf{S}_i\|_{\mathrm{F}}.$$

The scaling of each component of $\mathbf{S}$ by its Frobenius norm is simply to achieve even scaling of the components. The required square root can be obtained by symmetric eigendecomposition or pivoted Choleski decomposition. Now, using the factor $\boldsymbol{\mathcal{R}}$, from equation (9), and scaling it by its Frobenius norm, form a pivoted QR-decomposition

$$\begin{pmatrix} \boldsymbol{\mathcal{R}}/\|\boldsymbol{\mathcal{R}}\|_{\mathrm{F}} \\ \bar{\mathbf{E}}/\|\bar{\mathbf{E}}\|_{\mathrm{F}} \end{pmatrix} = \bar{\mathbf{Q}}\bar{\mathbf{R}}$$

and determine the rank r of the problem from the pivoted triangular factor $\bar{\mathbf{R}}$ (see Cline *et al.* (1979) and Golub and van Loan (1996)). The pivoting and rank determination indicates which parameters are unidentifiable (e.g. Golub and van Loan (1996), section 5.5).

3.4. Derivatives of $\hat{\boldsymbol{\beta}}$ with respect to the logarithmic smoothing parameters

The preceding Newton-based computation of the coefficients, $\hat{\boldsymbol{\beta}}$, leads to some moderately simple expressions for the derivatives of $\hat{\boldsymbol{\beta}}$ with respect to $\rho_j = \log(\lambda_j)$, which will be needed subsequently. Specifically

$$\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j} = -\exp(\rho_j)\mathbf{P}\mathbf{P}^{\mathrm{T}}\mathbf{S}_j\hat{\boldsymbol{\beta}}$$

and

$$\frac{\mathrm{d}^2\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j\,\mathrm{d}\rho_k} = \delta_j^k\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_k} - \mathbf{P}\mathbf{P}^{\mathrm{T}}\left\{\mathbf{X}^{\mathrm{T}}\mathbf{f}^{jk} + \exp(\rho_j)\mathbf{S}_j\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_k} + \exp(\rho_k)\mathbf{S}_k\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j}\right\}$$

where $\delta_j^k = 1$ if $j = k$ and $\delta_j^k = 0$ otherwise, and

$$f_i^{jk} = \frac{1}{2}\frac{\mathrm{d}\eta_i}{\mathrm{d}\rho_j}\frac{\mathrm{d}\eta_i}{\mathrm{d}\rho_k}\frac{\mathrm{d}w_i}{\mathrm{d}\eta_i},$$

$$\frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\rho_j} = \mathbf{X}\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j}.$$

Appendix C provides the derivation of these results, and Appendix D gives the expression for $\mathrm{d}w_i/\mathrm{d}\eta_i$. The leading order cost of these calculations is $O(M^2nq)$ where M is the number of smoothing parameters.

3.5. The rest of the restricted maximum likelihood objective and its derivatives

Given $\mathrm{d}\hat{\boldsymbol{\beta}}/\mathrm{d}\rho_j$ and $\mathrm{d}^2\hat{\boldsymbol{\beta}}/\mathrm{d}\rho_j\,\mathrm{d}\rho_k$ then the corresponding derivatives of $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ follow immediately. The derivatives of D with respect to $\boldsymbol{\rho}$ are then routine to calculate (see Wood (2008)

for full details). The remaining quantities in the REML (or ML) calculation are $|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S}|$, $\hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{S}\hat{\boldsymbol{\beta}}$ and the log-saturated-likelihood. These are covered here.

3.5.1. *Derivatives of log*$|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S}|$

Computation of $\log|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S}|$ itself was covered in Section 3.3. It will be stable provided that computations are conducted in the transformed space. The derivatives are also needed. Defining (with reference to Appendix D)

$$\mathbf{T}_j = \mathrm{diag}\left(\frac{1}{|w_i|}\frac{\partial w_i}{\partial \rho_j}\right),$$

$$\mathbf{T}_{jk} = \mathrm{diag}\left(\frac{1}{|w_i|}\frac{\partial^2 w_i}{\partial \rho_j \partial \rho_k}\right),$$

then some calculations using equations (16) and (17) from Appendix B show that

$$\frac{\partial \log|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S}|}{\partial \rho_k} = \mathrm{tr}(\mathbf{K}^{\mathrm{T}}\mathbf{T}_k\mathbf{K}) + \exp(\rho_k)\,\mathrm{tr}(\mathbf{P}^{\mathrm{T}}\mathbf{S}_k\mathbf{P})$$

and

$$\begin{aligned}\frac{\partial^2 \log|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}+\mathbf{S}|}{\partial \rho_k\,\partial \rho_j} = {} & \mathrm{tr}(\mathbf{K}^{\mathrm{T}}\mathbf{T}_{kj}\mathbf{K}) + \delta_k^j \exp(\rho_k)\,\mathrm{tr}(\mathbf{P}^{\mathrm{T}}\mathbf{S}_k\mathbf{P}) - \mathrm{tr}(\mathbf{K}^{\mathrm{T}}\mathbf{T}_k\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{T}_j\mathbf{K}) \\ & - \exp(\rho_j)\,\mathrm{tr}(\mathbf{K}^{\mathrm{T}}\mathbf{T}_k\mathbf{K}\mathbf{P}^{\mathrm{T}}\mathbf{S}_j\mathbf{P}) - \exp(\rho_k)\,\mathrm{tr}(\mathbf{K}^{\mathrm{T}}\mathbf{T}_j\mathbf{K}\mathbf{P}^{\mathrm{T}}\mathbf{S}_k\mathbf{P}) \\ & - \exp(\rho_k+\rho_j)\,\mathrm{tr}(\mathbf{P}^{\mathrm{T}}\mathbf{S}_k\mathbf{P}\mathbf{P}^{\mathrm{T}}\mathbf{S}_j\mathbf{P}).\end{aligned}$$

Although the **K**-, **P**- and the **T**-matrices all differ from those in Wood (2008), it is nonetheless possible to employ the tricks that are laid out in appendix C of Wood (2008) to evaluate the various traces in these expressions efficiently. The equivalent term for ML is slightly more involved and Appendix E provides details. Note that this step dominates the method's computational cost. The cost of second derivatives is $O(Mnq^2/2)$, whereas the cost of first derivatives is $O(nq^2)$ (the same as estimating $\boldsymbol{\beta}$). For large M, these costs suggest that quasi-Newton optimization, which only requires first derivatives, will sometimes be more efficient than full Newton optimization for optimization with respect to $\boldsymbol{\rho}$, although the fact that quasi-Newton optimization converges more slowly than Newton optimization complicates the comparison.

3.5.2. *Derivatives of* $\hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{S}\hat{\boldsymbol{\beta}}$

To complete the derivatives of D_p requires the derivatives of $\hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{S}\hat{\boldsymbol{\beta}}$. These are readily seen to be

$$\frac{\partial \hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{S}\hat{\boldsymbol{\beta}}}{\partial \rho_k} = 2\frac{\partial \hat{\boldsymbol{\beta}}^{\mathrm{T}}}{\partial \rho_k}\mathbf{S}\hat{\boldsymbol{\beta}} + \exp(\rho_k)\hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{S}_k\hat{\boldsymbol{\beta}}$$

and

$$\begin{aligned}\frac{\partial^2 \hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{S}\hat{\boldsymbol{\beta}}}{\partial \rho_k\,\partial \rho_j} = {} & 2\frac{\partial^2 \hat{\boldsymbol{\beta}}^{\mathrm{T}}}{\partial \rho_k\,\partial \rho_j}\mathbf{S}\hat{\boldsymbol{\beta}} + 2\frac{\partial \hat{\boldsymbol{\beta}}^{\mathrm{T}}}{\partial \rho_k}\mathbf{S}_j\hat{\boldsymbol{\beta}}\exp(\rho_j) + 2\frac{\partial \hat{\boldsymbol{\beta}}^{\mathrm{T}}}{\partial \rho_j}\mathbf{S}_k\hat{\boldsymbol{\beta}}\exp(\rho_k) + 2\frac{\partial \hat{\boldsymbol{\beta}}^{\mathrm{T}}}{\partial \rho_k}\mathbf{S}\frac{\partial \hat{\boldsymbol{\beta}}}{\partial \rho_j} \\ & + \delta_j^k \exp(\rho_k)\hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{S}_k\hat{\boldsymbol{\beta}},\end{aligned}$$

which have $O(M^2q^2)$ computational cost.

3.5.3. Scale-parameter-related derivatives

For known scale parameter cases, all the derivatives that are required for direct Newton optimization of the REML or ML criteria have now been obtained. However, when ϕ is unknown some further work is still needed (the dependence on ϕ has none of the exploitable linearity of the dependence on λ_i, which is why it must be treated separately).

If $\phi = \exp(\rho_\phi)$ is estimated by direct REML then we need only

$$-\frac{\partial l_r}{\partial \rho_\phi} = -\frac{D_p}{2\phi} - l_s'(\phi)\phi - \frac{M_p}{2},$$

$$-\frac{\partial^2 l_r}{\partial \rho_\phi^2} = \frac{D_p}{2\phi} - l_s''(\phi)\phi^2 - l_s'(\phi)\phi,$$

$$-\frac{\partial^2 l_r}{\partial \rho_\phi \partial \rho_k} = -\frac{1}{2\phi}\frac{\partial D_p}{\partial \rho_k}$$

and the derivatives of l_r with respect to $\boldsymbol{\rho}$. (These derivatives also serve to emphasize that direct estimation works only with full likelihood, not quasi-likelihood.)

If $\hat{\phi}$ is the Pearson statistic over $n - M_p$, where M_p is the penalty null space dimension (the number of fixed effects), then an alternative version of the REML score and its derivatives is

$$-\hat{l}_r = \frac{D_p}{2\hat{\phi}} - l_s(\hat{\phi}) + K - \frac{M_p}{2}\log(2\pi\hat{\phi}),$$

$$-\frac{\partial \hat{l}_r}{\partial \rho_k} = \frac{\partial D_p}{\partial \rho_k}\frac{1}{2\hat{\phi}} - \left\{\frac{D_p}{2\hat{\phi}^2} + l_s'(\hat{\phi}) + \frac{M_p}{2\hat{\phi}}\right\}\frac{\partial \hat{\phi}}{\partial \rho_k} + \frac{\partial K}{\partial \rho_k},$$

and

$$-\frac{\partial^2 \hat{l}_r}{\partial \rho_k \partial \rho_j} = \frac{\partial^2 D_p}{\partial \rho_k \partial \rho_j}\frac{1}{2\hat{\phi}} - \left(\frac{\partial D_p}{\partial \rho_k}\frac{\partial \hat{\phi}}{\partial \rho_j} + \frac{\partial D_p}{\partial \rho_j}\frac{\partial \hat{\phi}}{\partial \rho_k}\right)\frac{1}{2\hat{\phi}^2} + \left\{\frac{D_p}{\hat{\phi}^3} - l_s''(\hat{\phi}) + \frac{M_p}{2\hat{\phi}^2}\right\}\frac{\partial \hat{\phi}}{\partial \rho_k}\frac{\partial \hat{\phi}}{\partial \rho_j}$$

$$-\left\{\frac{D_p}{2\phi^2} + l_s'(\hat{\phi}) + \frac{M_p}{2\hat{\phi}}\right\}\frac{\partial^2 \hat{\phi}}{\partial \rho_k \partial \rho_j} + \frac{\partial^2 K}{\partial \rho_k \partial \rho_j}.$$

These require the derivatives of $\hat{\phi}$, which are easily obtained from the known derivatives of $\hat{\boldsymbol{\beta}}$ with respect to the smoothing parameters, combined with the derivatives of the Pearson statistic, which are given in Appendix F.

The ML derivative expressions are identical to those given in this subsection, if we set $M_p = 0$ (for ML, the fixed effects are not integrated out, and in consequence the direct dependence on the number of fixed effects goes). Whichever version of REML or ML is used, derivatives of the saturated log-likelihood with respect to ϕ are required: Appendix G gives some common examples.

3.6. Other smoothness selection criteria

Although it was not possible to adapt the Wood (2008) method to optimize REML or ML reliably, the method that is proposed here can readily optimize prediction error criteria of the sort that were discussed in Wood (2008). In fact the new method has the advantage of eliminating a potential difficulty with the Wood (2008) method, namely that, when using a non-canonical link

in the presence of outliers, the Fisher-based PIRLS could (rarely) require step length reduction at convergence, which could cause the subsequent derivative iterations to fail.

Prediction error criteria are based on the the deviance, Pearson statistic and effective degrees of freedom of the model, formally defined as tr($\mathbf{F}$) where

$$\mathbf{F} = (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}.$$

Clearly the methods that have been described so far deal with the deviance and Pearson statistic, but the derivatives of tr($\mathbf{F}$) require some more work. The results of this are provided in Appendix H. There are good reasons for preferring $\mathbf{W}$ to be based on the Fisher weights in the computation of $\mathbf{F}$. Doing so guarantees that both $\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S}$ and $\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}$ are positive definite, which ensures that the effective degrees of freedom are well defined. There are also robustness-to-outlier arguments (e.g. Demidenko (2004)) for using the Fisher weights for constructing variance estimates, despite the general superiority of observed information over expected information for this purpose (Efron and Hinkley, 1978).

4. Some simulation comparisons

The REML- and ML-based methods, which are proposed here, were compared with GCV (AIC for known scale parameters) and PQL (based on the version that is implemented in R function `glmmPQL`; Venables and Ripley (2002)), as means for selecting smoothing parameters. For each replicate, 400 data y_i were simulated (independently) from an exponential family distribution, with mean μ_i where

$$g(\mu_i)/k = f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}).$$

g is a known link function and the x_{ji} are independent identically distributed (IID) uniform on $(0, 1)$. k is used to control the signal-to-noise ratio. The f_j are plotted in Fig. 2(f). Five distribution–link combinations were used, with 200 replicates performed for each: normal–identity, gamma–log-, Tweedie–log- (variance power 1.5), binary–logit and Poisson–log-link. For each case k was set to achieve a squared correlation coefficient between μ_i and y_i of about 0.5. A generalized additive model (GAM) with the correct link–error structure was fitted to each replicate, but with the linear predictor given by a sum of smooth functions of the three actual predictors plus a smooth function of a nuisance predictor, which was IID uniform, but did not influence the true μ_i. The four-component smooth models were represented by rank 10 thin plate regression splines (Wood, 2003), except for the third component, for which a rank of 30 was used. Smoothing parameters were chosen by each of REML, ML, PQL and GCV (or AIC when the scale parameter was known), for each replicate.

Model performance was judged by calculating the mean-square error (MSE) in reconstructing the true linear predictor, at the observed covariate values. In the case of binary data, this measure is rather unstable for fitted probabilities in the vicinity of 0 or 1, so the probability scale was used in place of the linear predictor scale.

The results are summarized in Fig. 2. Boxplots show the distributions, over 200 replicates, of differences in MSE between each alternative method and REML. Before plotting, the MSEs are divided by the MSE for REML estimation, averaged over the the case being plotted. In all cases a Wilcoxon signed rank test indicates that REML has lower MSE than the competing method (p-value less than 10^{-3} except for the PQL–ML comparison for the Tweedie distribution where $p = 0.04$). The Tweedie variance power was 1.5. PQL failed in 16, 10, 22 and seven replicates, for the gamma, Tweedie, binary and Poisson data respectively. The other methods converged successfully for every replicate. The most dramatic difference is between REML and PQL for

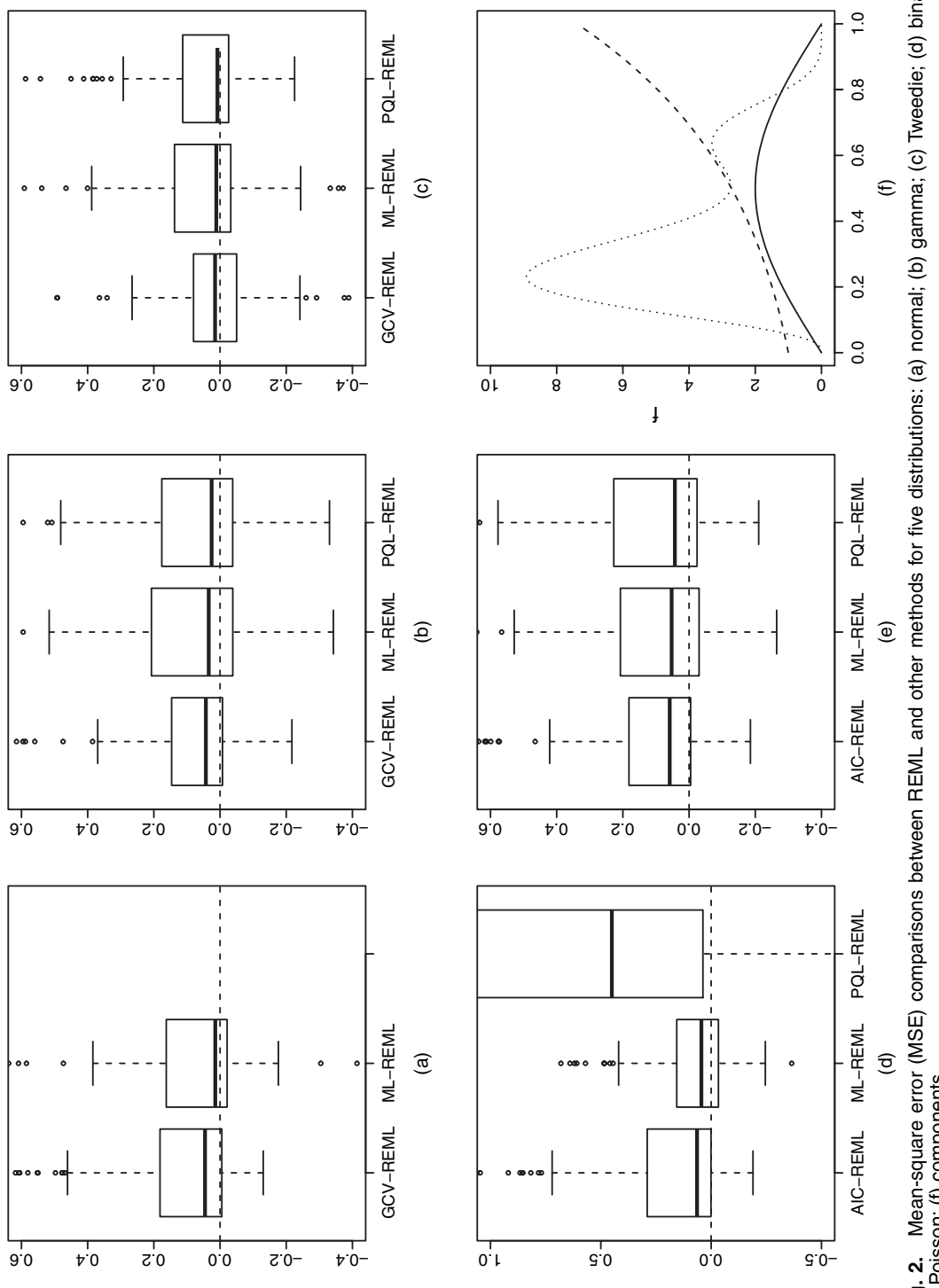

Fig. 2. Mean-square error (MSE) comparisons between REML and other methods for five distributions: (a) normal; (b) gamma; (c) Tweedie; (d) binary; (e) Poisson; (f) components

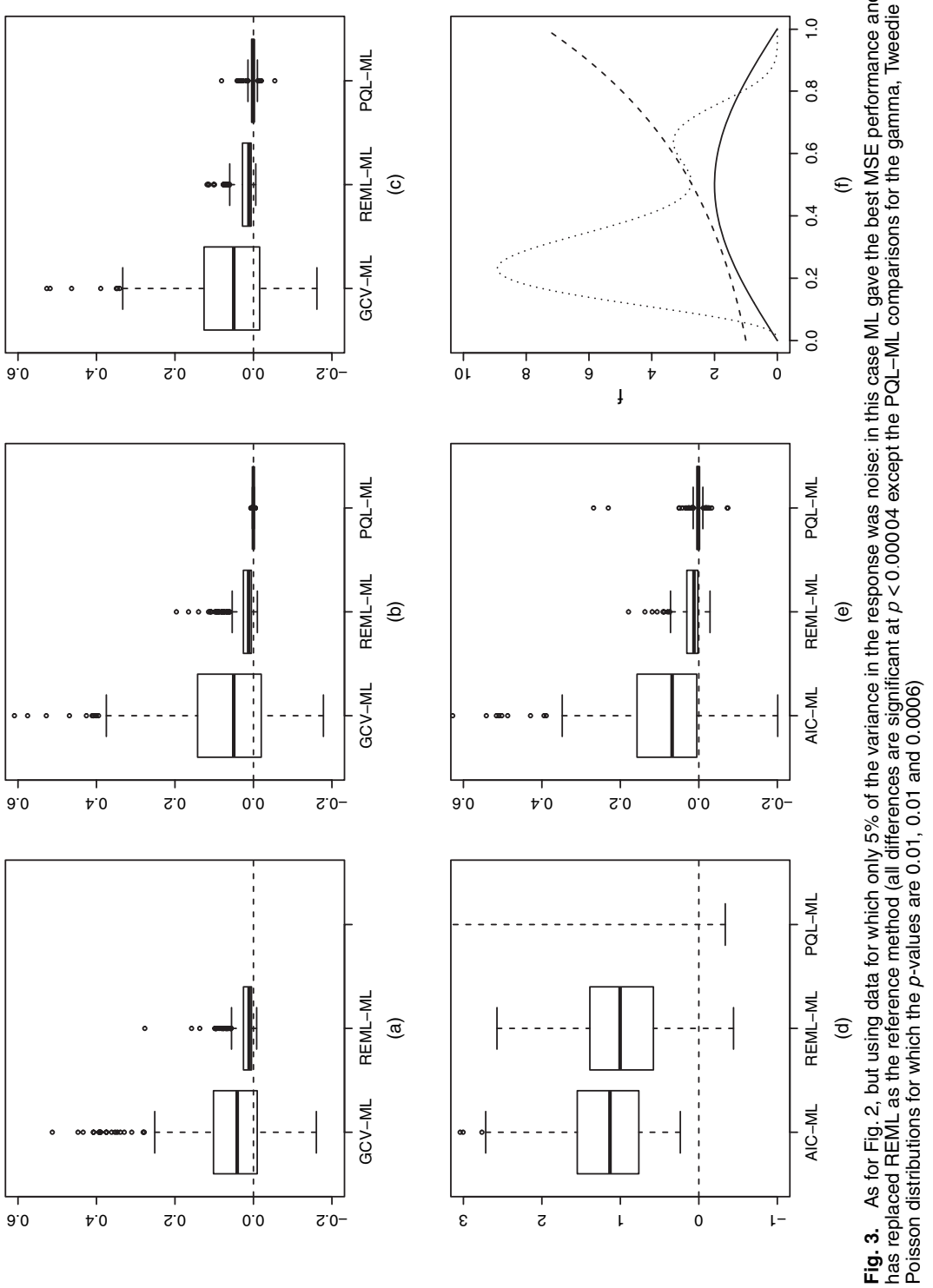

Fig. 3. As for Fig. 2, but using data for which only 5% of the variance in the response was noise: in this case ML gave the best MSE performance and so has replaced REML as the reference method (all differences are significant at $p < 0.00004$ except the PQL–ML comparisons for the gamma, Tweedie and Poisson distributions for which the p-values are 0.01, 0.01 and 0.0006)

binary data, where PQL has a substantial tail of poor fits, reflecting the well-known fact that PQL is poor for binary data. Note also the skew in the GCV–REML comparisons: this seems to result from a smallish proportion of GCV- or AIC-based replicates substantially overfitting. The mean time per replicate for GCV or AIC, REML and ML was about 0.7 s on a 1.33-GHz Intel U7700 computer running LINUX (on a mid-range laptop). PQL took between 10 and 20 times longer. All computations were performed with R 2.9.2 (R Development Core Team, 2008) and R package `mgcv` version 1.6-1 (which includes a Tweedie family based on Dunn and Smith (2005)).

The experiment was repeated at lower noise levels: first for noise levels such that the r^2-value between μ_i and y_i was about 0.7 and then for still lower noise levels so that the r^2-value was about 0.95. Fig. 3 shows the results for the lowest noise level. In this case ML gives the best MSE performance, although REML is not much worse and still better than the prediction error criteria. The intermediate noise level results are not shown, but indicate ML and REML to be almost indistinguishable, and both better than prediction error criteria. It seems likely that the superiority of ML over REML in the lowest noise case relates to Wahba's (1985) demonstration that REML undersmooths, asymptotically: ML will of course smooth more but is still consistent (Kauermann *et al.*, 2009). Similarly the failure of prediction error methods to show any appreciable catch-up as noise levels were reduced, despite their asymptotic superiority in MSE terms, presumably relates to the excruciatingly slow convergence rates for prediction-criteria-based estimates, obtained in Härdle *et al.* (1988).

The two problematic examples from the introduction to Wood (2008), Figs 1 and 2, were also repeated with the methods that are developed here: convergence was unproblematic and reasonable fits were obtained. See Appendix A for some further comparisons with another alternative method.

The simulation evidence supports the implication of Reiss and Ogden's (2009) work, that REML (and hence the structurally very similar ML) may have practical advantages over GCV or AIC for smoothing parameter selection, and reinforces the message from Wood (2008), that direct nested optimization is quicker and more reliable than selecting smoothing parameters on the basis of approximate working models.

5. Examples

This section presents three example applications which, as special cases of penalized GLMs, are straightforward given the general method that is proposed in this paper.

5.1. Simple P-spline adaptive smoothing

An important feature of the method proposed is that it is stable even when different penalties act on intersecting sets of parameters. Tensor product smooths that are used for smooth interaction terms are an obvious important case where this occurs (see for example Wood (2006), section 4.1.8), but adaptive smoothing provides a less-well-known example, as illustrated in this section, using adaptive P-splines.

The 'P-splines' of Eilers and Marx (1996) combine B-spline basis functions and discrete penalties on the basis coefficients, to obtain flexible spline-like smoothers. For example, if we let $b_j(x)$ denote B-spline basis functions, with evenly spaced knots, then an unknown function f can be represented (approximately) as

$$f(x) = \sum_{i=1}^{K} \beta_i b_i(x)$$

and the wiggliness of this function can be measured by using the discrete penalty

$$\mathcal{P}_{\text{ordinary}} = \sum_{i=2}^{K-1} (\beta_{i-1} - 2\beta_i + \beta_{i+1})^2,$$

or higher or lower order alternatives. The penalty can be used as a smoothing penalty in fitting. One of the reasons that *P*-splines have proved so popular is the ease with which they can be modified to perform non-standard smoothing tasks, at relatively little loss of performance relative to more computationally complex smoothers. Adaptive smoothing illustrates this.

An adaptive penalty is easily constructed by allowing the terms in the penalty to have different weights, depending on i, and hence on x. For example:

$$\mathcal{P} = \sum_{i=2}^{K-1} c_i(\beta_{i-1} - 2\beta_i + \beta_{i+1})^2.$$

Now defining $d_i = \beta_{i-1} - 2\beta_i + \beta_{i+1}$, and $\mathbf{D}$ to be the matrix of coefficients such that $\mathbf{d} = \mathbf{D}\boldsymbol{\beta}$, we have

$$\mathcal{P} = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\,\mathrm{diag}(\mathbf{c})\mathbf{D}\boldsymbol{\beta}.$$

The elements c_i are unknown, but we could use a *B*-spline basis to model the c_i as a smooth function of i or x so that $\mathbf{c} = \mathbf{C}\boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ is a vector of unknown (positive) coefficients. In this case

$$\mathcal{P} = \sum_j \lambda_j \boldsymbol{\beta}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\,\mathrm{diag}(\mathbf{C}_{.,j})\mathbf{D}\boldsymbol{\beta}$$

where $\mathbf{C}_{.,j}$ is column j of $\mathbf{C}$, i.e. the adaptive penalty has become a sum of penalties multiplied by smoothing parameters λ_j. The same construction can be used for smooths of several covariates, using tensor products of *P*-splines. See Krivobokova *et al.* (2008) for a more sophisticated *P*-spline-based approach to this problem.

The obvious advantage of the approach that is given here is that it allows adaptive smoothers to be used as components of penalized GLMs in the same way as any other smooth. As an example consider smoothing the well-known motorcycle crash data that were used in Silverman (1985). The response a_i is acceleration of the head of a test dummy in a simulated motorcycle crash, and it depends on time t_i. A simple model is

$$a_i = f(t_i) + \varepsilon_i$$

where the ε_i are IID $N(0, \sigma^2)$ (although a better model would have σ^2 depending on time as well). Given that the data show a low acceleration phase followed by rapid changes in acceleration followed by a smooth return to zero, it is possible to make the case that the degree of penalization of f should depend on t. A model was therefore fitted in which f was represented by using a rank 40 cubic *B*-spline basis (even knot spacing), penalized by using the adaptive penalty given above, $\boldsymbol{\lambda}$ having dimension 5 (although the results are rather insensitive to the exact choice here). The smoothing parameters $\boldsymbol{\lambda}$ were chosen by REML.

The results are shown in Fig. 4, which also includes a fit in which a single-penalty rank 40 thin plate regression spline is used to represent $f(t)$. The single-penalty case must use the same degree of penalization for all t, with the result that the curve at low and high times appears underpenalized and too bumpy, presumably to accommodate the high degree of variability at

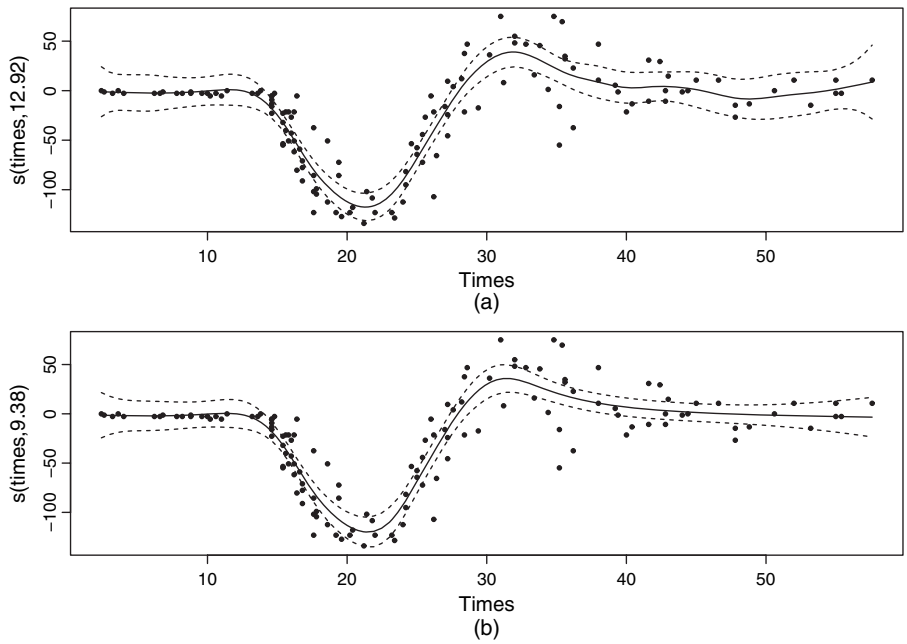

Fig. 4. Two attempts to smooth the motorcycle crash data (all smoothing parameters were chosen by REML; note that the adaptive smoother uses fewer effective degrees of freedom and produces a fit which appears to show better adaptation to the data): (a) the smooth as a rank 40 penalized thin plate regression spline; (b) a simple adaptive smoother of the type discussed in Section 5.1

intermediate times. The adaptive fit took 1.3 s, compared with 0.15 s for the single-penalty fit (see Section 4, for computer details).

5.2. Generalized regression of scalars on functions

The fact that the method that is described in this paper has been developed for the rather general model (1) means that it can be used for models that superficially appear to be rather different from a GAM. To illustrate this, this section revisits an example from Reiss and Ogden (2009) but makes use of the new method to employ a more general model than theirs, based on non-Gaussian errors with multiple penalties.

Consider a response y_i which is dependent on predictor function $z_i(x)$, where x may be univariate or multivariate. In this case an appropriate model might be

$$g(\mu_i) = \alpha + \int f(x)\, z_i(x)\, \mathrm{d}x, \tag{15}$$

with y_i an observation from some exponential family distribution, with mean μ_i. $f(x)$ is an unknown 'coefficient' function and must be estimated. It is straightforward to extend the model by adding other smooth terms to the linear predictor (the right-hand side). In practice the integral will be approximated by quadrature, with the midpoint rule being adequate in most cases. Suppose that the domain of $z_i(x)$ is finite and let x_j denote points at which z_i has been observed (with even spacing h). The model becomes

$$g(\mu_i) = \alpha + h \sum_j f(x_j)\, z_i(x_j).$$

Any penalized regression spline basis can be used for f, and model estimation proceeds as for any other penalized GLM. For more detail on such models see Marx and Eilers (1999), Escabias *et al.* (2004), Ramsay and Silverman (2005) or Reiss and Ogden (2007) (and also Wahba (1990)).

As an example, consider trying to predict the octane rating of gasoline (or petrol) from its near infrared spectrum. For internal combustion engines in which a fuel–air mixture is compressed within the cylinders before combustion, it is important that the fuel–air mixture does not spontaneously ignite owing to compressive heating. Such early combustion results in 'knocking' and poor engine performance. The octane rating of fuel measures its resistance to knocking. It is a somewhat indirect measure: the lowest compression ratio at which the fuel causes knocking is recorded. The octane rating is the percentage of iso-octane in the mixture of n-heptane and iso-octane with the same lowest knocking compression ratio as the fuel sample. Measuring octane rating requires special variable compression test engines, and it would be rather simpler to measure the octane from spectral measurements on a fuel sample, if this were possible.

Fig. 5(a) shows near infrared spectra for 60 gasoline samples (from Kalivas (1997), as provided by Wehrens and Mevik (2007)). The octane rating of each sample has also been measured. Model (15) is a possibility for such data (where y_i is octane rating, $z_i(x)$ is the ith spectrum and x is wavelength). The octane rating is positive and continuous (at least in theory), and there is some indication of increasing variance with mean (see Fig. 5(c)), so a gamma distribution with log-link is an appropriate initial model. The spectra themselves are rather spiky, with some smooth regions interspersed with regions of very rapid variation. It seems sensible to allow the coefficient function $f(x)$ the possibility of behaving in a similar way, so representing f by using the same sort of adaptive smooth as was used in the previous section is appropriate. Estimation of this model is then just a case of estimating a GLM subject to multiple penalization. The remaining panels of Fig. 5 show the results of this fitting, with REML smoothness selection.

Note that the coefficient function appears to be contrasting the two peak regions with the trough between them, with the extreme ends of the spectra apparently adding little. The model explains around 98% of the deviance in octane rating, and the residual plots look plausible (including a QQ-plot of deviance residuals, which is not shown).

5.3. Generalized additive model term selection and null space penalties

Smoothing parameter selection does most of the work in selecting between models of differing complexity, but it does not usually remove a term from the model altogether. If the smoothing parameter for a term tends to ∞, this usually causes the term to tend towards some simple, but non-zero, function of its covariate. For example, as its smoothing parameter tends to ∞, a cubic regression spline term will tend to a straight line. It seems logical to decide on whether or not terms should be included in the model by using the same criterion as used for smoothness selection, but how should this be achieved in practice? Tutz and Binder (2006) proposed one solution to the model selection problem, by using a boosting approach to perform fitting, smoothness selection and term selection simultaneously. They also provided evidence that in very data poor settings, with many spurious covariates, this approach can be much better than the alternatives. This section proposes a possible alternative to boosting, in which each smooth term is given an extra penalty, which will shrink to zero any functions that are in the null space of the usual penalty.

For example, consider a smooth with K coefficients $\boldsymbol{\beta}$ and penalty matrix $\mathcal{S}$, with null space dimension M_s, so that the wiggliness penalty is $\boldsymbol{\beta}^{\mathrm{T}}\mathcal{S}\boldsymbol{\beta}$. Now consider the eigendecomposition $\mathcal{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}$. The first $K - M_s$ eigenvalues Λ_i will be positive, and the last M_s will be 0. Writing

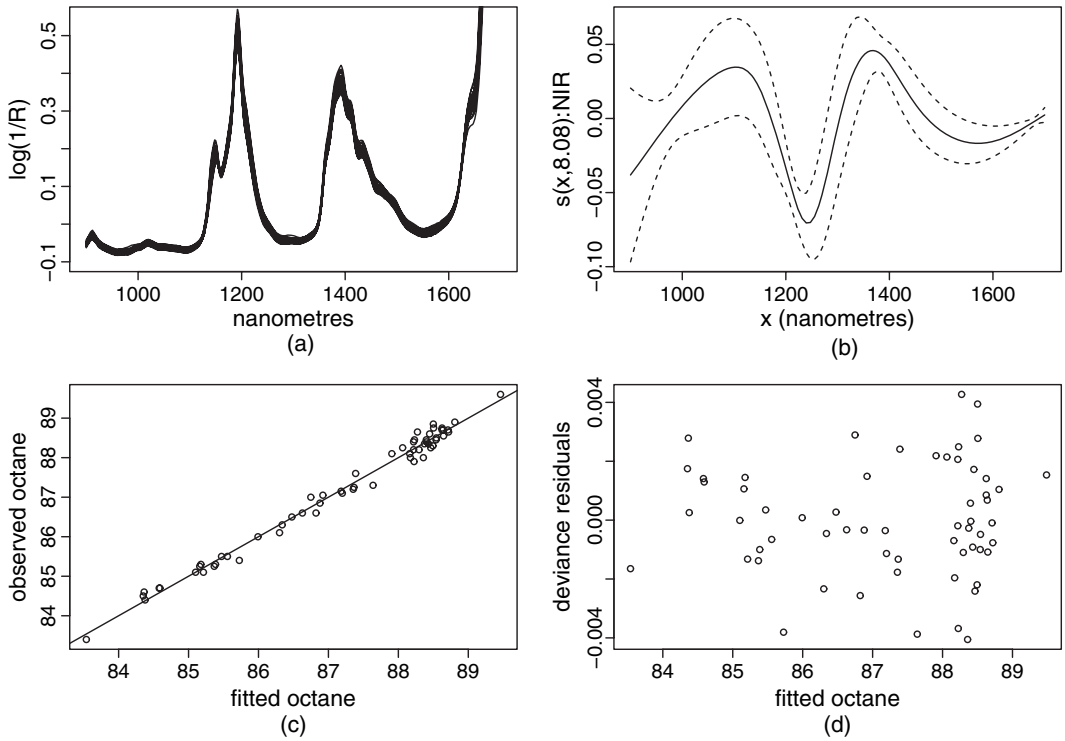

Fig. 5. (a) Near infrared spectra for 60 samples of gasoline (the *y*-axis is the logarithm of the inverse of reflectance, which is measured every 2 nm; these spectra ought to be able to predict the octane rating of the samples; the spectra actually reach 1.2 at the right-hand end but, since this region turns out to have little predictive power, the *y*-axis has been truncated to show more detail at lower wavelengths); (b) estimated coefficient function for the model given in Section 5.2, with factor *h* absorbed (the inner product of this with the spectrum for a sample gives the predicted octane rating); (c) observed *versus* fitted ratings; (d) deviance residuals for the model *versus* fitted octane rating

$\mathbf{\Lambda}_+$ for the $(K - M_s) \times (K - M_s)$ diagonal matrix containing only the positive eigenvalues, and $\mathbf{U}_+$ for the $K \times (K - M_s)$ matrix of corresponding eigenvectors, then $\mathcal{S} = \mathbf{U}_+ \mathbf{\Lambda}_+ \mathbf{U}_+^{\mathrm{T}}$. Now let $\mathbf{U}_-$ be the $K \times M_s$ matrix of the eigenvectors corresponding to zero eigenvalues. $\mathbf{U}_+$ forms a basis for the space of coefficients corresponding to the 'wiggly' component of the smooth, whereas $\mathbf{U}_-$ is a basis for the components of zero wiggliness—the null space of the penalty. The two bases are orthogonal. So, if we want to produce a penalty which penalizes only the null space of the penalty, we could use $\boldsymbol{\beta}^{\mathrm{T}} \mathcal{S}_N \boldsymbol{\beta}$ where $\mathcal{S}_N = \mathbf{U}_- \mathbf{U}_-^{\mathrm{T}}$. If a smooth term is already subject to multiple penalties (e.g. a tensor product smooth or an adaptive smooth), the same basic construction holds, but the null space is obtained from the eigendecomposition of the sum of the original penalty matrices. Note that this construction is general and completely automatic.

This sort of construction could be used with any smoothing parameter selection method, not just REML or ML, but it is less appealing if used with a method which is prone to under-smoothing, as GCV seems to be.

As a small example, Poisson data were simulated assuming a log-link and a linear predictor made up of the sum of the three functions shown in Fig. 2(f) applied to three sets of 200 IID $U(0, 1)$ covariates. Six more IID $U(0, 1)$ nuisance covariates were simulated. A GAM was fitted to the simulated data, assuming a Poisson distribution and log-link, and with a linear predictor

consisting of a sum of nine smooth functions of the nine covariates. Each smooth function was represented by using a rank 10 cubic regression spline (actually *P*-splines for GAM boosting). The model was fitted by using four different methods: the GAM boosting method of Tutz and Binder (2006), using version 1.1 of R package `GAMBoost` (with penalty set to 500 to ensure that each fit used well over the 50 boosting steps that were suggested as the minimum by Tutz and Binder (2006)); GCV smoothness selection, with the null space penalties that were suggested here, REML with no null space penalties and REML with null space penalties. 200 replicates of this experiment were run, and the MSE in the linear predictor at the covariate values was recorded for each method for each replicate.

Fig. 6 shows the results. REML with null space penalties achieves lower MSE than REML without null space penalties, and substantially better performance than GCV with null space penalties or GAM boosting. The success of the methods in identifying which components should be in the model at all was also recorded. For GAM boosting the methods given in the `GAMBoost` package were employed, whereas, for the null space penalties, terms with effective degrees of freedom greater than 0.2 were deemed to have been selected. On this basis the false negative

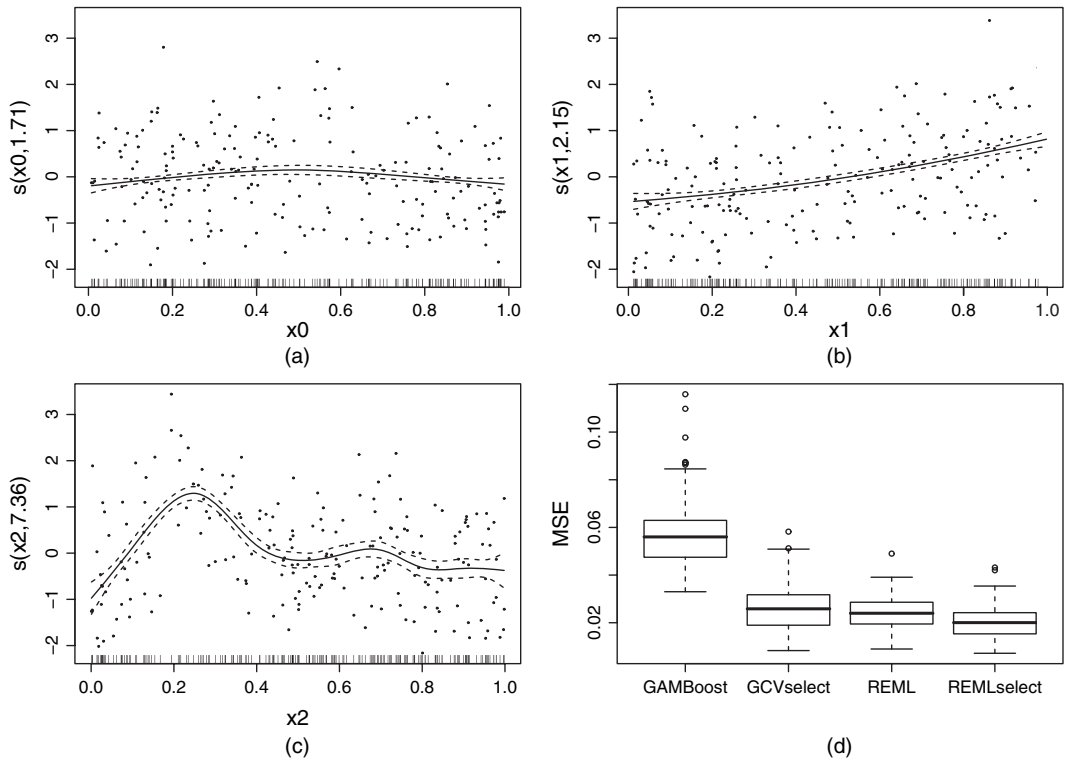

Fig. 6. Model selection example (models were fitted to Poisson data simulated from a linear predictor made up of the three terms shown in Fig. 2(f); the linear predictors of the fitted models also included smooth functions of six additional nuisance predictors; four alternative fitting methods were used for each replicate simulation): (a)–(c) typical estimates of the terms that actually made up the true linear predictor, using REML, with selection penalties (partial Pearson residuals are shown for each smooth estimate); (d) distribution, over 200 replicates, of the MSE of the models fitted by each of the methods ('GAMBoost' is fitted by using Tutz and Binder's (2006) boosting method, 'GCVselect' is for models with selection penalties under GCV smoothness selection, 'REML' is REML smoothness selection without selection penalties and 'REMLselect' is for REML smoothness selection with selection penalties)

selection rates (rates at which influential covariates were not selected) were 0.6% for boosting and 0.16% for the other methods. The false positive selection rates (rates at which spurious terms were selected) were 67%, 71% and 62% for boosting, GCV and REML respectively. REML with null space penalties took just under 6 s per fit, on average, whereas boosting took about 2.5 min per fit. Note that the example here has relatively high information content, relative to the scenarios that were investigated by Binder and Tutz (2006): with less information boosting is still appealing.

6. Discussion

The method that was proposed in this paper offers a general computationally efficient way of estimating the smoothing parameters of models of the form (1), when the f_j are represented by using penalized regression splines and the coefficients $\boldsymbol{\beta}$ are estimated by optimizing expression (3). With this method, REML- or ML-based estimation of semiparametric GLMs can rival the estimation of ordinary parametric GLMs for routine computational reliability. Previously such efficiency and reliability were only available for prediction error criteria, such as GCV. This means that the advantages of REML or ML estimation that were outlined in Section 1.1 need no longer be balanced against the more reliable fitting methods that are available for GCV or AIC. The cost of this enhancement is that the method proposed has a somewhat more complex mathematical structure than the previous prediction-error-based methods (e.g. Wood (2008)), but since the method is freely available in R package `mgcv` (from version 1.5) this is not an obstacle to its use.

Given that REML or ML estimation requires that we view model (1) as a generalized linear mixed model, then an obvious question is why should it be treated as a special case for estimation purposes, rather than estimated by general generalized linear mixed model software? The answer lies in the special nature of the λ_i. The fact that they enter the penalty or precision matrix linearly, facilitates both the evaluation of derivatives to computational accuracy and the ability to stabilize the computations via the method of Appendix B. In addition the λ_i are unusual precision parameters in that their 'true' value is often infinite. This behaviour can cause problems for general purpose methods, which cannot exploit the advantages of the linear structure. Conversely, the method that is proposed here can be used to fit any generalized linear mixed model where the precision matrix is a linear combination of known matrices but, since it is not designed to exploit the sparse structure that many random effects have, it may not be the most efficient method for so doing.

A limitation of the method that was presented here is that it is designed to be efficient when the f_j are represented by using penalized regression splines as described in Wahba (1980), Parker and Rice (1985), Eilers and Marx (1996), Marx and Eilers (1998), Ruppert *et al.* (2003), Wood (2003) etc. These 'intermediate rank' smoothers have become very popular over the last decade, as researchers realized that many of the advantages of splines could be obtained without the computational expense of full splines: an opinion which turns out to be well founded theoretically (see Gu and Kim (2002), Hall and Opsomer (2005) and Kauermann *et al.* (2009)). But, despite its wide applicability, the penalized regression spline approach has limitations. The most obvious is that relatively low rank smooths are unsuitable for modelling short-range auto-correlation (particularly spatial). Where this deficiency matters, Rue *et al.* (2009) offer an attractive alternative approach, by directly estimating additive smooth components of the linear predictor, with very sparse $\mathbf{S}_j$-matrices directly penalizing these components. The required sparsity can be obtained by modelling the smooth components as Markov random fields of some sort. Provided that the number of smoothing parameters is quite low, then the methods offer very

efficient computation for this class of problem, as well as better inferences about the smoothing parameters themselves. When the model includes large numbers of random effects, but not all components have the sparsity that is required by Rue *et al.* (2009), or when the number of smoothing parameters or variance parameters is moderate to large, then the simulation-based Bayesian approach of Fahrmeir, Lang and co-workers (e.g. Lang and Brezger (2004), Brezger and Lang (2006) and Fahrmeir and Lang (2001)) is likely to be more efficient than the method that is proposed here, albeit applicable to a more restricted range of penalized GLMs, because of restrictions on the $\mathbf{S}_j$ that are required to maintain computational efficiency.

An interesting area for further work would be to establish relative convergence rates for the $\hat{f}_j$ under REML, ML and GCV smoothness selection. It is not difficult to arrange for $\hat{f}_j$ to be consistent under either approach, at least when spline-like bases are used for the f_j in model (1). Without penalization, all that we require is that the basis dimension grows with sample size n sufficiently fast that the spline approximation error declines at a faster rate than the sampling variance of $\hat{f}_j$, but sufficiently slow that $\dim(\boldsymbol{\beta})/n \to 0$ (so that the observed likelihood converges to its expectation). This is not difficult to achieve, given the good approximation theoretic properties of splines. If smoothing parameters are chosen to be sufficiently small, then penalization will *reduce* the MSE at any n, so consistency can be maintained under penalized estimation. In fact, asymptotically, GCV *minimizes* the MSE (or a generalized equivalent), so the $\hat{f}_j$ will be consistent under GCV estimation. Since REML smooths less than GCV, asymptotically (Wahba, 1985), then the same must hold for REML. However, establishing the relative convergence *rates* that are actually achieved under the two alternatives appears to be more involved.

Acknowledgements

I am especially grateful to Mark Bravington for explanation of the implicit function theorem, and for providing the original fisheries modelling examples that broke Fisher-scoring-based approaches, and to Phil Reiss for first alerting me to the real practical advantages of REML and providing an early preprint of Reiss and Ogden (2009). The Commonwealth Scientific and Industrial Research Organisation paid for a visit to Hobart where the Tweedie work was done, and it became clear that the Wood (2008) method could not be extended to REML. Thanks also go to Mark Bravington, Merrilee Hurn and Alistair Spence for some helpful discussions during the painful process that led to the Section 3.1–Appendix B method. Phil Reiss, Mark Bravington, Nicole Augustin and Giampiero Marra all provided very helpful comments on an earlier draft of this paper. The Joint Editor, Associate Editor and two referees all made suggestions which substantially improved the paper, for which I am also grateful.

Appendix A: Convergence failures of previous restricted maximum likelihood schemes

Wood (2008) provides some examples of convergence failure for the PQL approach, in which smoothing parameters are estimated iteratively by REML or ML estimation of working linear mixed models. The alternative scheme that is proposed in the literature has been implemented by Brezger *et al.* (2007) in the BayesX package. Like PQL, this scheme need not converge (as Brezger *et al.* (2007) explicitly pointed out), but Brezger *et al.* (2007) employed an ingenious heuristic stabilization trick which seems to lead to superior performance to that of PQL in this regard. However, it is not difficult to find realistic examples that still give convergence problems. For example the following code was used in R version 2.7.1 to generate data with a relatively benign collinearity problem and a mild mean–variance relationship problem:

```
set.seed(1);n<-1000;alpha<-.75
x0<-runif(n);x1<-x0*alpha+(1-alpha)*runif(n)
x2<-runif(n);x3<-x2*alpha+(1-alpha)*runif(n)
x4<-runif(n);x5<-runif(n)
f0<-function(x)2*sin(pi*x)
f1<-function(x)exp(2*x)
f2<-function(x)0.2*x^11*(10*(1-x))^6+10*(10*x)^3*(1-x)^10
f<-f0(x0)+f1(x1)+f2(x2)
y<-rgamma(f,exp(f/4),scale=1.2)
```

Fitting the model

$$\log\{\mathbb{E}(y_i)\} = f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}) + f_4(x_{4i}) + f_5(x_{5i}) + f_6(x_{6i}),$$

$y_i \sim$ gamma, in BayesX version 1.5.0, representing each f by a (default) rank 20 P-spline, resulted in convergence failure, with the estimates zigzagging without ever converging. Nine subsequent replicates of this experiment yielded two more convergence failures of the same sort, three catastrophic divergences and four problem-free convergences (although one of these took more than 200 iterations). Fitting the same model to these data sets by using the methods that are proposed in this paper gave no problems and sensible function reconstructions in each case.

Appendix B: $\left|\sum_i \lambda_i \mathbf{S}_i\right|_+$

As discussed in Section 3.1, a stable method for calculating $\log(|\Sigma_i \lambda_i \mathbf{S}_i|_+)$ and its derivatives with respect to $\rho_i = \log(\lambda_i)$ is required, when the λ_i may be wildly different in magnitude. This appendix provides such a method by extending the simple approach that was described in Section 3.1.

Here it is assumed that $q \times q$ matrix $\mathbf{S} = \Sigma_i \lambda_i \mathbf{S}_i$ is formally of full rank. When this is not so then the following initial transformation will be required. First form the symmetric eigendecomposition:

$$\tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^{\mathrm{T}} = \sum_i \mathbf{S}_i / \|\mathbf{S}_i\|_{\mathrm{F}},$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. Now let $\mathbf{U}_+$ denote the columns of $\tilde{\mathbf{U}}$ corresponding to positive eigenvalues. The transformation $\tilde{\mathbf{S}}_i = \mathbf{U}_+^{\mathrm{T}} \mathbf{S}_i \mathbf{U}_+$ is then applied and the methods of this appendix are utilized on the transformed matrices. It is easy to show that $|\mathbf{S}|_+ = |\Sigma_i \lambda_i \tilde{\mathbf{S}}_i|$, and that $\Sigma_i \lambda_i \tilde{\mathbf{S}}_i$ has full rank. For the rest of this appendix it is assumed that this transformation has been applied if necessary, and the tildes are dropped.

Initialization: set $K = 0$, $Q = q$ and $\bar{\mathbf{S}}_i = \mathbf{S}_i, \forall i$. Set $\gamma = \{1 \ldots M\}$, where M is the number of $\mathbf{S}_i$-matrices.

Similarity transformation: the following steps are iterated until the termination criterion is met (at step 4).

Step 1: set $\Omega_i = \|\bar{\mathbf{S}}_i\|_{\mathrm{F}} \lambda_i, \forall i \in \gamma$.

Step 2: create $\alpha = \{i : \Omega_i \geqslant \varepsilon \max(\Omega_i), i \in \gamma\}$ and $\gamma' = \{i : \Omega_i < \varepsilon \max(\Omega_i), i \in \gamma\}$ where ε is, for example, the cube root of the machine precision. So α indexes the dominant terms out of those remaining.

Step 3: find the eigenvalues of $\Sigma_{i\in\alpha} \bar{\mathbf{S}}_i / \|\bar{\mathbf{S}}_i\|_{\mathrm{F}}$ and use these to determine the formal rank r of any summation of the form $\Sigma_{i\in\alpha} \lambda_i \bar{\mathbf{S}}_i$ where the λ_i are positive. The rank is determined by counting the number of eigenvalues that are larger than ε times the dominant eigenvalue. ε is typically the machine precision raised to a power in [0.7,0.9].

Step 4: if $r = Q$ then terminate. The current $\mathbf{S}$ is the $\mathbf{S}$ to use for determinant calculation.

Step 5: find the eigendecomposition $\mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}} = \Sigma_{i\in\alpha} \lambda_i \bar{\mathbf{S}}_i$, where the eigenvalues are arranged in descending order on the leading diagonal of $\mathbf{D}$. Let $\mathbf{U}_r$ be the first r columns of $\mathbf{U}$ and $\mathbf{U}_n$ the remaining columns.

Step 6: write $\mathbf{S}$ in partitioned form

$$\mathbf{S} = \begin{pmatrix} \mathbf{A}_{K\times K} & \mathbf{B}_{K\times Q} \\ \mathbf{B}^{\mathrm{T}}_{Q\times K} & \mathbf{C}_{Q\times Q} \end{pmatrix}$$

where the subscripts denote dimensions (rows × columns). Then set $\mathbf{B}' = \mathbf{B}\mathbf{U}$ and

$$\mathbf{C}' = \begin{pmatrix} \mathbf{D}_r + \mathbf{U}_r^{\mathrm{T}} \mathbf{S}_{\gamma'} \mathbf{U}_r & \mathbf{U}_r^{\mathrm{T}} \mathbf{S}_{\gamma'} \mathbf{U}_n \\ \mathbf{U}_n^{\mathrm{T}} \mathbf{S}_{\gamma'} \mathbf{U}_r & \mathbf{U}_n^{\mathrm{T}} \mathbf{S}_{\gamma'} \mathbf{U}_n \end{pmatrix}$$

where $\mathbf{S}_{\gamma'} = \Sigma_{i\in\gamma'} \lambda_i \bar{\mathbf{S}}_i$. Then

$$\mathbf{S}' = \begin{pmatrix} \mathbf{I}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{U}^{\mathrm{T}} \end{pmatrix} \mathbf{S} \begin{pmatrix} \mathbf{I}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{U} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B}' \\ \mathbf{B}'^{\mathrm{T}} & \mathbf{C}' \end{pmatrix}$$

and $|\mathbf{S}| = |\mathbf{S}'|$. The key point here is that the effect of the terms that are indexed by α has been concentrated into an $r \times r$ block, with rows and columns to the lower right of that block uncontaminated by 'large machine 0s' from the terms indexed by α.

Step 7: define

$$\mathbf{T}_\alpha = \begin{pmatrix} \mathbf{I}_K & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix},$$

$$\mathbf{T}_{\gamma'} = \begin{pmatrix} \mathbf{I}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{U} \end{pmatrix}$$

and transform

$$\mathbf{S}_i \leftarrow \mathbf{T}_\alpha^{\mathrm{T}} \mathbf{S}_i \mathbf{T}_\alpha \qquad \forall i \in \alpha$$

and

$$\mathbf{S}_i \leftarrow \mathbf{T}_{\gamma'}^{\mathrm{T}} \mathbf{S}_i \mathbf{T}_{\gamma'} \qquad \forall i \in \gamma'.$$

These transformations facilitate derivative calculations using the transformed $\mathbf{S}$.

Step 8: transform $\bar{\mathbf{S}}_i \leftarrow \mathbf{U}_n^{\mathrm{T}} \bar{\mathbf{S}}_i \mathbf{U}_n, \forall i \in \gamma'$.

Step 9: set $K \leftarrow K + r$, $Q \leftarrow Q - r$, $\mathbf{S} \leftarrow \mathbf{S}'$ and $\gamma \leftarrow \gamma'$. Return to step 1.

Note that the orthogonal matrix which similarity transforms the original $\mathbf{S}$ to the final transformed version can be accumulated as the algorithm progresses, to produce the $\mathbf{Q}_s$ of Section 3.1.

The effect of the preceding iteration is to concentrate the dominant terms in $\mathbf{S}$ into the smallest possible block of leftmost columns, with these terms having no effect beyond those columns. Next the most dominant terms in the remainder are concentrated in the smallest possible number of immediately succeeding columns, again with no effect to the right of these columns. This pattern is repeated. Since QR-decomposition operates on columns of $\mathbf{S}$, without mixing columns, it can now be used to evaluate stably the determinant of the transformed $\mathbf{S}$. Alternative methods of determinant calculation (e.g. Choleski or symmetric eigendecomposition) would require an additional preconditioning step.

It is straightforward to obtain a stable matrix square root of the transformed $\mathbf{S}$, which maintains the column separation that is evident in $\mathbf{S}$ itself. Defining diagonal matrix $P_{ii} = |S_{ii}|^{1/2}$, form the Choleski factor of the diagonally preconditioned version of $\mathbf{S}$, i.e.

$$\mathbf{L}\mathbf{L}^{\mathrm{T}} = \mathbf{P}^{-1}\mathbf{S}\mathbf{P}^{-1}.$$

Then $\mathbf{E} = \mathbf{L}^{\mathrm{T}}\mathbf{P}$ is a matrix square root, such that $\mathbf{E}^{\mathrm{T}}\mathbf{E} = \mathbf{S}$. Preconditioning is essential to ensure that the square root is computable without ever requiring numerical truncation, since the latter would cause spurious discontinuous changes in the numerical value of $|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S}|$, which depends on $\mathbf{E}$.

Finally, note that, on the basis of the general results,

$$\frac{\partial \log |\mathbf{F}|}{\partial x_j} = \mathrm{tr}\left(\mathbf{F}^{-1} \frac{\partial \mathbf{F}}{\partial x_j} \right) \tag{16}$$

and

$$\frac{\partial^2 \log |\mathbf{F}|}{\partial x_i \, \partial x_j} = \mathrm{tr}\left(\mathbf{F}^{-1} \frac{\partial^2 \mathbf{F}}{\partial x_i \, \partial x_j} \right) - \mathrm{tr}\left(\mathbf{F}^{-1} \frac{\partial \mathbf{F}}{\partial x_i} \mathbf{F}^{-1} \frac{\partial \mathbf{F}}{\partial x_j} \right) \tag{17}$$

(see Harville (1997)), the expressions for the derivatives are as follows (all right-hand side terms are transformed versions):

$$\frac{\partial \log |\mathbf{S}|}{\partial \rho_j} = \lambda_j \, \mathrm{tr}(\mathbf{S}^{-1}\mathbf{S}_j)$$

and

$$\frac{\partial^2 \log |\mathbf{S}|}{\partial \rho_i \, \partial \rho_j} = \delta_j^i \lambda_i \, \mathrm{tr}(\mathbf{S}^{-1}\mathbf{S}_i) - \lambda_i \lambda_j \, \mathrm{tr}(\mathbf{S}^{-1}\mathbf{S}_i\mathbf{S}^{-1}\mathbf{S}_j).$$

Appendix C: Derivatives of $\hat{\boldsymbol{\beta}}$ by using implicit differentiation

When full Newton optimization is used in place of Fisher scoring to obtain $\hat{\boldsymbol{\beta}}$, then there is no computational advantage in iterating for the derivatives of $\hat{\boldsymbol{\beta}}$ with respect to $\boldsymbol{\rho}$ (as in Wood (2008)), rather than

exploiting the implicit function theorem to obtain them directly by implicit differentiation. This is because Newton-based PIRLS requires exactly the same quantities as implicit differentiation. This appendix provides the details.

Define

$$D_p = D(\boldsymbol{\beta}) + \sum_m \exp(\rho_m)\boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}_m\boldsymbol{\beta},$$

and note that in this appendix some care must be taken to distinguish total derivatives of D_p, which encompass all variability with respect to a variable, as opposed to partial derivatives of the expression for D_p, which ignore dependence of $\hat{\boldsymbol{\beta}}$ on $\boldsymbol{\rho}$.

C.1. Partial derivatives of D_p

$$\frac{\partial D}{\partial \beta_r} = -2\sum_i \omega_i \frac{y-\mu_i}{V(\mu_i)\,g'(\mu_i)}\mathbf{X}_{ir},$$

$$\frac{\mathrm{d}\mu_i}{\mathrm{d}\beta_r} = \frac{X_{ir}}{g'(\mu_i)},$$

from which it follows (after some calculation) that

$$\frac{\partial^2 D}{\partial \beta_r\,\partial \beta_m} = \sum_i 2w_i X_{im} X_{ir}$$

where w_i is the Newton version. Consequently

$$\frac{\partial^3 D}{\partial \beta_r\,\partial \beta_m\,\partial \beta_l} = \sum_i \frac{\mathrm{d}w_i}{\mathrm{d}\eta_i} X_{im} X_{ir} X_{il}.$$

Note that the *partials* of D with respect to $\boldsymbol{\rho}$ are 0.

Turning to $P = \Sigma_m \exp(\rho_m)\boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}_m\boldsymbol{\beta}$ (so $D_p = D + P$) we have

$$\nabla_\beta P = 2\sum_m \exp(\rho_m)\mathbf{S}_m\boldsymbol{\beta},$$

$$\nabla^2_\beta P = 2\sum_m \exp(\rho_m)\mathbf{S}_m.$$

Furthermore

$$\frac{\partial \nabla_{\boldsymbol{\beta}} P}{\partial \rho_j} = 2\exp(\rho_j)\mathbf{S}_j\boldsymbol{\beta},$$

$$\frac{\partial^2 \nabla_\beta P}{\partial \rho_j\,\partial \rho_k} = 2\delta^k_j \exp(\rho_j)\mathbf{S}_j\boldsymbol{\beta},$$

$$\frac{\partial \nabla^2_\beta P}{\partial \rho_j} = 2\exp(\rho_j)\mathbf{S}_j.$$

C.2. Derivatives of $\hat{\boldsymbol{\beta}}$ with respect to $\boldsymbol{\rho}$

$\hat{\boldsymbol{\beta}}$ is the solution to

$$\frac{\mathrm{d}D_p}{\mathrm{d}\beta_r} = 0.$$

Since this equation always holds at $\hat{\boldsymbol{\beta}}$, we have

$$\frac{\mathrm{d}^2 D_p}{\mathrm{d}\beta_r\,\mathrm{d}\rho_j} = \sum_m \frac{\partial^2 D_p}{\partial \beta_r\,\partial \beta_m}\frac{\mathrm{d}\beta_m}{\mathrm{d}\rho_j} + \frac{\partial^2 D_p}{\partial \beta_r\,\partial \rho_j} = 0,$$

at $\hat{\boldsymbol{\beta}}$, i.e.

$$\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j} = -\left(\frac{\partial^2 D_p}{\partial\boldsymbol{\beta}\,\partial\boldsymbol{\beta}^{\mathrm{T}}}\right)^{-1} \frac{\partial\nabla_\beta D_p}{\partial\rho_j}.$$

Differentiating again we obtain

$$\frac{\mathrm{d}^3 D_p}{\mathrm{d}\beta_r\,\mathrm{d}\rho_j\,\mathrm{d}\rho_k} = \sum_l\sum_m \frac{\partial^3 D_p}{\partial\beta_r\,\partial\beta_m\,\partial\beta_l}\frac{\mathrm{d}\beta_m}{\mathrm{d}\rho_j}\frac{\mathrm{d}\beta_l}{\mathrm{d}\rho_k} + \sum_m \frac{\partial^3 D_p}{\partial\beta_r\,\partial\beta_m\,\partial\rho_k}\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j} + \sum_m \frac{\partial^2 D_p}{\partial\beta_r\,\partial\beta_m}\frac{\mathrm{d}^2\beta_m}{\mathrm{d}\rho_j\,\mathrm{d}\rho_k}$$
$$+ \sum_m \frac{\partial^3 D_p}{\partial\beta_r\,\partial\beta_m\,\partial\rho_j}\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_k} + \frac{\partial^3 D_p}{\partial\beta_r\,\partial\rho_j\,\partial\rho_k} = 0.$$

Now

$$\frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\rho_j} = \mathbf{X}\frac{\mathrm{d}\boldsymbol{\beta}}{\mathrm{d}\rho_j},$$

so using the expression for the third partial of D/D_p with respect to $\boldsymbol{\rho}$ and rearranging we obtain

$$\frac{\mathrm{d}^2\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j\,\mathrm{d}\rho_k} = -\left(\frac{\partial^2 D_p}{\partial\boldsymbol{\beta}\,\partial\boldsymbol{\beta}^{\mathrm{T}}}\right)^{-1}\left\{\frac{\partial^2\nabla_\beta D_p}{\partial\rho_j\,\partial\rho_k} + \mathbf{X}^{\mathrm{T}}\mathbf{f}^{jk} + 2\exp(\rho_j)\mathbf{S}_j\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_k} + 2\exp(\rho_k)\mathbf{S}_k\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j}\right\}$$
$$= \delta_j^k\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_k} - \left(\frac{\partial^2 D_p}{\partial\boldsymbol{\beta}\,\partial\boldsymbol{\beta}^{\mathrm{T}}}\right)^{-1}\left\{\mathbf{X}^{\mathrm{T}}\mathbf{f}^{jk} + 2\exp(\rho_j)\mathbf{S}_j\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_k} + 2\exp(\rho_k)\mathbf{S}_k\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j}\right\}$$

where

$$f_i^{jk} = \frac{\mathrm{d}\eta_i}{\mathrm{d}\rho_j}\frac{\mathrm{d}\eta_i}{\mathrm{d}\rho_k}\frac{\mathrm{d}w_i}{\mathrm{d}\eta_i}.$$

The inverse required is $\mathbf{P}\mathbf{P}^{\mathrm{T}}/2$ (with derivatives of dropped parameters set to 0 by this choice).

Appendix D: Derivatives of w

In this appendix primes denote differentiation with respect to μ_i. First the derivatives of α_i are useful:

$$\alpha_i' = -\left(\frac{V_i'}{V_i} + \frac{g_i''}{g_i'}\right) + (y_i - \mu_i)\left(\frac{V_i''}{V_i} - \frac{V_i'^2}{V_i^2} + \frac{g_i'''}{g_i'} - \frac{g_i''^2}{g_i'^2}\right)$$

and

$$\alpha_i'' = -2\left(\frac{V_i''}{V_i} - \frac{V_i'^2}{V_i^2} + \frac{g_i'''}{g_i'} - \frac{g_i''^2}{g_i'^2}\right) + (y_i - \mu_i)\left(\frac{V_i'''}{V_i} - \frac{3V_i'V_i''}{V_i^2} + \frac{2V_i'^3}{V_i^3} + \frac{g_i''''}{g_i'} - \frac{3g_i'''g_i''}{g_i'^2} + \frac{2g_i''^3}{g_i'^3}\right).$$

The key derivatives of w_i are then

$$\frac{\mathrm{d}w_i}{\mathrm{d}\eta_i} = \frac{w_i}{g_i'}\left(\frac{\alpha_i'}{\alpha_i} - \frac{V_i'}{V_i} - 2\frac{g_i''}{g_i'}\right)$$

and

$$\frac{\mathrm{d}^2 w_i}{\mathrm{d}\eta_i^2} = \frac{1}{w_i}\left(\frac{\mathrm{d}w_i}{\mathrm{d}\eta_i}\right)^2 - \frac{\mathrm{d}w_i}{\mathrm{d}\eta_i}\frac{g_i''}{g_i'^2} + \frac{w_i}{g_i'^2}\left(\frac{\alpha_i''}{\alpha_i} - \frac{\alpha_i'^2}{\alpha_i^2} - \frac{V_i''}{V_i} + \frac{V_i'^2}{V_i^2} - 2\frac{g'''}{g_i'} + 2\frac{g_i''^2}{g_i'^2}\right).$$

The derivatives of $\boldsymbol{\eta}$ with respect to $\boldsymbol{\rho}$ are obtained from the derivatives of $\hat{\boldsymbol{\beta}}$ with respect to $\boldsymbol{\rho}$, so the derivatives of w_i with respect to $\boldsymbol{\rho}$ follow easily. Note that setting $\alpha_i \equiv 1$, and its derivatives to 0, recovers Fisher scoring.

Appendix E: Marginal likelihood determinant term and derivatives

ML requires computation of $\log|\bar{\mathbf{X}}^{\mathrm{T}}\mathbf{W}\bar{\mathbf{X}} + \bar{\mathbf{S}}|$ and its derivatives (see Section 2.1). This requires further work. First note that explicit formation and decomposition of $\sqrt{\mathbf{W}}\mathbf{X}\mathbf{U}_1$ would be wasteful. All that is needed is the (pivoted) QR-decomposition

$$\mathbf{R}\mathbf{U}_1 = \bar{\mathbf{Q}}\bar{\mathbf{R}}$$

where $\mathbf{R}$ is from Section 3.3. $\mathbf{R}$ (and $\mathbf{Q}_1$) should not be truncated here, even if there is rank deficiency: instead $\bar{\mathbf{R}}$ and $\bar{\mathbf{Q}}$ should be. It is then easy to show that

$$\bar{\mathbf{X}}^{\mathrm{T}}\mathbf{W}\bar{\mathbf{X}} + \bar{\mathbf{S}} = \bar{\mathbf{R}}^{\mathrm{T}}(\mathbf{I} - 2\bar{\mathbf{Q}}^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}}\mathbf{I}^{-}\mathbf{Q}_1\bar{\mathbf{Q}})\bar{\mathbf{R}}.$$

Forming the singular value decomposition

$$\mathbf{I}^{-}\mathbf{Q}_1\bar{\mathbf{Q}} = \bar{\mathbf{U}}\bar{\mathbf{D}}\bar{\mathbf{V}}^{\mathrm{T}},$$

define

$$\bar{\mathbf{P}} = \begin{pmatrix} \bar{\mathbf{R}}^{-1}\bar{\mathbf{V}}(\mathbf{I} - 2\bar{\mathbf{D}}^2)^{-1/2} \\ \mathbf{0} \end{pmatrix},$$

$$\bar{\mathbf{K}} = \mathbf{Q}_1\bar{\mathbf{Q}}\bar{\mathbf{V}}(\mathbf{I} - 2\bar{\mathbf{D}}^2)^{-1/2}.$$

Then $|\bar{\mathbf{X}}^{\mathrm{T}}\mathbf{W}\bar{\mathbf{X}} + \bar{\mathbf{S}}| = |\bar{\mathbf{R}}|^2|\mathbf{I} - 2\bar{\mathbf{D}}^2|$ and the expressions for the derivatives of $\log|\bar{\mathbf{X}}^{\mathrm{T}}\mathbf{W}\bar{\mathbf{X}} + \bar{\mathbf{S}}|$ are as in Section 3.5.1, but with $\bar{\mathbf{P}}$ and $\bar{\mathbf{K}}$ in place of $\mathbf{P}$ and $\mathbf{K}$ and the $\mathbf{S}_k$ replaced by $\bar{\mathbf{S}}_k = \mathbf{U}_1^{\mathrm{T}}\mathbf{S}_k\mathbf{U}_1$ (pivoted in the same way as the $\bar{\mathbf{R}}$).

Appendix F: Pearson statistic

The derivatives of the Pearson statistic with respect to the coefficients are required. Wood (2008) provided these in a form which holds only under Fisher scoring. Here is the general form.

$$P = \sum_i P_i \qquad P_i = \frac{\omega_i(y_i - \mu_i)^2}{V_i}.$$

So we need

$$\frac{\mathrm{d}P_i}{\mathrm{d}\beta_j} = \frac{\mathrm{d}P_i}{\mathrm{d}\eta_i}X_{ij},$$

$$\frac{\mathrm{d}^2P_i}{\mathrm{d}\beta_j\,\mathrm{d}\beta_k} = \frac{\mathrm{d}^2P_i}{\mathrm{d}\eta_i^2}X_{ij}X_{ik}.$$

The requisite derivatives are

$$\frac{\mathrm{d}P_i}{\mathrm{d}\eta_i} = -\frac{1}{g_i'}\left\{\frac{2\omega_i(y_i - \mu_i)}{V_i} + P_i\frac{V_i'}{V_i}\right\}$$

and

$$\frac{\mathrm{d}^2P_i}{\mathrm{d}\eta_i^2} = \frac{g_i''}{g_i'^3}\left\{\frac{2\omega_i(y_i - \mu_i)}{V_i} + P_i\frac{V_i'}{V_i}\right\} + \frac{1}{g_i'^2}\left\{\frac{2\omega_i}{V_i} + \frac{2\omega_i(y_i - \mu_i)}{V_i}\frac{V_i'}{V_i} - g_i'\frac{\mathrm{d}P_i}{\mathrm{d}\eta_i}\frac{V_i'}{V_i} - P_i\left(\frac{V_i''}{V_i} - \frac{V_i'^2}{V_i^2}\right)\right\}.$$

Appendix G: Derivatives of the saturated log-likelihood

When the scale parameter is fixed and known, as in the binomial and Poisson cases, then l_s is irrelevant and its derivative with respect to ϕ is 0. Otherwise l_s and derivatives are needed. Here are three common examples.

(a) *Gaussian*:

$$l_s = -\log(\phi)/2 - \log(2\pi)/2,$$
$$l_s' = -1/2\phi,$$
$$l_s'' = 1/2\phi^2.$$

(b) *Inverse Gaussian*:

$$l_s = -\log(\phi)/2 - \log(2\pi y^3)/2,$$
$$l_s' = -1/2\phi,$$
$$l_s'' = 1/2\phi^2.$$

(c) *Gamma*:

$$l_s = -\log\Gamma(1/\phi) - \frac{\log(\phi)}{\phi} - \frac{1}{\phi} - \log(y).$$

Writing $\log\Gamma$ to mean the log-gamma function (to be differentiated as a whole):

$$l'_s = \frac{\log\Gamma'(1/\phi)}{\phi^2} + \frac{\log(\phi)}{\phi^2},$$
$$l''_s = -\frac{\log\Gamma''(1/\phi)}{\phi^4} - \frac{2\log\Gamma'(1/\phi)}{\phi^3} + \frac{1-2\log(\phi)}{\phi^3}.$$

The `lgamma`, `digamma` and `trigamma` functions in R evaluate $\log\Gamma$, $\log\Gamma'$ and $\log\Gamma''$ respectively.

Appendix H: Derivatives of tr(F)

Prediction error criteria, such as GCV, involve the effective degrees of freedom of a model defined as $\mathrm{tr}(\mathbf{F})$ where

$$\mathbf{F} = (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}.$$

To optimize such criteria by using the method that was developed here requires differentiation of $\mathrm{tr}(\mathbf{F})$ with respect to the logarithmic smoothing parameters. Define $\mathbf{G} = \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S}$. Note that $\mathbf{G}^{-1}\mathbf{X}^{\mathrm{T}}\sqrt{\mathbf{W}} = \mathbf{P}\mathbf{K}^{\mathrm{T}}$, $\sqrt{\mathbf{W}}\mathbf{X}\mathbf{G}^{-1}\mathbf{X}^{\mathrm{T}}\sqrt{\mathbf{W}} = \mathbf{K}\mathbf{K}^{\mathrm{T}}$ and $\mathbf{G}^{-1} = \mathbf{P}\mathbf{P}^{\mathrm{T}}$. Also define $\mathbf{T}_j$ and $\mathbf{T}_{jk}$ as in Section 3.5.1 (and *not* as in Wood (2008)), and diagonal matrix $\mathbf{I}^+$ where $I^+_{ii} = -1$ if $w_i < 0$ and $I^+_{ii} = 1$ otherwise. Now $\mathbf{F} = \mathbf{P}\mathbf{K}^{\mathrm{T}}\mathbf{I}^+\sqrt{\mathbf{W}}\mathbf{X}$ and

$$\frac{\partial \mathbf{F}}{\partial \rho_j} = -\mathbf{G}^{-1}\left(\mathbf{X}^{\mathrm{T}}\frac{\partial \mathbf{W}}{\partial \rho_j}\mathbf{X} + \exp(\rho_j)\mathbf{S}_j\right)\mathbf{G}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{G}^{-1}\mathbf{X}^{\mathrm{T}}\frac{\partial \mathbf{W}}{\partial \rho_j}\mathbf{X},$$

so that

$$\frac{\partial \mathrm{tr}(\mathbf{F})}{\partial \rho_j} = -\mathrm{tr}(\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{T}_j\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{I}^+) - \exp(\rho_j)\,\mathrm{tr}(\mathbf{K}\mathbf{P}^{\mathrm{T}}\mathbf{S}_j\mathbf{P}\mathbf{K}^{\mathrm{T}}\mathbf{I}^+) + \mathrm{tr}(\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{T}_j).$$

Second derivatives are more tedious:

$$\begin{aligned}\frac{\partial^2 \mathbf{F}}{\partial \rho_j\,\partial \rho_k} = &\left[\mathbf{G}^{-1}\left(\mathbf{X}^{\mathrm{T}}\frac{\partial \mathbf{W}}{\partial \rho_j}\mathbf{X} + \exp(\rho_j)\mathbf{S}_j\right)\mathbf{G}^{-1}\left(\mathbf{X}^{\mathrm{T}}\frac{\partial \mathbf{W}}{\partial \rho_k}\mathbf{X} + \exp(\rho_k)\mathbf{S}_k\right)\mathbf{G}^{-1}\right]^{\ddagger}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}\\ &- \mathbf{G}^{-1}\left(\mathbf{X}^{\mathrm{T}}\frac{\partial^2 \mathbf{W}}{\partial \rho_j\,\partial \rho_k}\mathbf{X} + \delta^k_j\exp(\rho_j)\mathbf{S}_j\right)\mathbf{G}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} - \mathbf{G}^{-1}\left(\mathbf{X}^{\mathrm{T}}\frac{\partial \mathbf{W}}{\partial \rho_j}\mathbf{X} + \exp(\rho_j)\mathbf{S}_j\right)\mathbf{G}^{-1}\mathbf{X}^{\mathrm{T}}\frac{\partial \mathbf{W}}{\partial \rho_k}\mathbf{X}\\ &- \mathbf{G}^{-1}\left(\mathbf{X}^{\mathrm{T}}\frac{\partial \mathbf{W}}{\partial \rho_k}\mathbf{X} + \exp(\rho_k)\mathbf{S}_k\right)\mathbf{G}^{-1}\mathbf{X}^{\mathrm{T}}\frac{\partial \mathbf{W}}{\partial \rho_j}\mathbf{X} + \mathbf{G}^{-1}\mathbf{X}^{\mathrm{T}}\frac{\partial^2 \mathbf{W}}{\partial \rho_j\partial \rho_k}\mathbf{X},\end{aligned}$$

where $[\mathbf{A}]^{\ddagger} = \mathbf{A} + \mathbf{A}^{\mathrm{T}}$. It follows that

$$\begin{aligned}\frac{\partial^2 \mathrm{tr}(\mathbf{F})}{\partial \rho_j\,\partial \rho_k} = &\,2\,\mathrm{tr}(\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{T}_k\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{T}_j\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{I}^+) + 2\,\exp(\rho_j)\,\mathrm{tr}(\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{T}_k\mathbf{K}\mathbf{P}^{\mathrm{T}}\mathbf{S}_j\mathbf{P}\mathbf{K}^{\mathrm{T}}\mathbf{I}^+)\\ &+ 2\,\exp(\rho_k)\,\mathrm{tr}(\mathbf{K}\mathbf{P}^{\mathrm{T}}\mathbf{S}_k\mathbf{P}\mathbf{K}^{\mathrm{T}}\mathbf{T}_j\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{I}^+) + 2\,\exp(\rho_k+\rho_j)\,\mathrm{tr}(\mathbf{K}\mathbf{P}^{\mathrm{T}}\mathbf{S}_k\mathbf{P}\mathbf{P}^{\mathrm{T}}\mathbf{S}_j\mathbf{P}\mathbf{K}^{\mathrm{T}}\mathbf{I}^+)\\ &- \mathrm{tr}(\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{T}_{jk}\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{I}^+) - \delta^k_j\exp(\rho_j)\,\mathrm{tr}(\mathbf{K}\mathbf{P}^{\mathrm{T}}\mathbf{S}_j\mathbf{P}\mathbf{K}^{\mathrm{T}}\mathbf{I}^+) - 2\,\mathrm{tr}(\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{T}_k\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{T}_j)\\ &- \exp(\rho_j)\,\mathrm{tr}(\mathbf{K}\mathbf{P}^{\mathrm{T}}\mathbf{S}_j\mathbf{P}\mathbf{K}^{\mathrm{T}}\mathbf{T}_k) - \exp(\rho_k)\,\mathrm{tr}(\mathbf{K}\mathbf{P}^{\mathrm{T}}\mathbf{S}_k\mathbf{P}\mathbf{K}^{\mathrm{T}}\mathbf{T}_j) + \mathrm{tr}(\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{T}_{jk}).\end{aligned}$$

Although the **K**-, **P**- and **T**-matrices are all different from those in Wood (2008), and the $\mathbf{I}^+$-matrices did not feature there at all, it is still possible to use the tricks that are listed in appendix C of Wood (2008) to evaluate these terms efficiently, with only minor adjustment.

There is a strong argument for employing Fisher-scoring-based weights in place of Newton-based weights in the definition of **F**. This requires redefining **W**, $\mathbf{T}_k$ and $\mathbf{T}_{jk}$ and setting $\mathbf{I}^+$ to **I**, but otherwise the computations are identical. This change removes the possibility of $\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}$ having negative eigenvalues, which can occasionally lead to nonsensical computed effective degrees of freedom.

References

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Donngarra, J., Du Croz, J., Greenbaum, A., Hammerling, S., McKenney, A. and Sorenson, D. (1999) *LAPACK Users' Guide*, 3rd edn. Philadelphia: Society for Industrial and Applied Mathematics.

Anderssen, R. S. and Bloomfield, P. (1974) A time series approach to numerical differentiation. *Technometrics*, **16**, 69–75.

Breslow, N. E. and Clayton, D. G. (1993) Approximate inference in generalized linear mixed models. *J. Am. Statist. Ass.*, **88**, 9–25.

Brezger, A., Kneib, T. and Lang, S. (2007) BayesX 1.5.0. University of Munich, Munich. (Available from `http://www.stat.uni-muenchen.de/~bayesx`.)

Brezger, A. and Lang, S. (2006) Generalized structured additive regression based on Bayesian P-splines. *Computnl Statist. Data Anal.*, **50**, 967–991.

Cline, A. K., Moler, C. B., Stewart, G. W. and Wilkinson, J. H. (1979) An estimate for the condition number of a matrix. *SIAM J. Numer. Anal.*, **13**, 293–309.

Craven, P. and Wahba, G. (1979) Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross validation. *Numer. Math.*, **31**, 377–403.

Davison, A. C. (2003) *Statistical Models*. Cambridge: Cambridge University Press.

Demidenko, E. (2004) *Mixed Models: Theory and Applications*. Hoboken: Wiley.

Dunn, P. K. and Smith, G. K. (2005) Series evaluation of Tweedie exponential dispersion model densities. *Statist. Comput.*, **15**, 267–280.

Efron, B. and Hinkley, D. V. (1978) Assessing the accuracy of the maximum likelihood estimator: observed versus expected Fisher information. *Biometrika*, **65**, 457–487.

Eilers, P. H. C. and Marx, B. D. (1996) Flexible smoothing with B-splines and penalties. *Statist. Sci.*, **11**, 89–121.

Eilers, P. H. C. and Marx, B. D. (2002) Generalized linear additive smooth structures. *J. Computnl Graph. Statist.*, **11**, 758–783.

Escabias, M., Aguilera, A. M. and Valderrama, M. J. (2004) Principal component estimation of functional logistic regression: discussion of two different approaches. *Nonparam. Statist.*, **16**, 365–384.

Fahrmeir, L., Kneib, T. and Lang, S. (2004) Penalized structured additive regression for space time data: a Bayesian perspective. *Statist. Sin.*, **14**, 731–761.

Fahrmeir, L. and Lang, S. (2001) Bayesian inference for generalized additive mixed models based on Markov random field priors. *Appl. Statist.*, **50**, 201–220.

Golub, G. H. and van Loan, C. F. (1996) *Matrix Computations*, 3rd edn. Baltimore: Johns Hopkins University Press.

Green, P. J. and Silverman, B. W. (1994) *Nonparametric Regression and Generalized Linear Models*. London: Chapman and Hall.

Gu, C. (1992) Cross validating non-Gaussian data. *J. Computnl Graph. Statist.*, **1**, 169–179.

Gu, C. (2002) *Smoothing Spline ANOVA Models*. New York: Springer.

Gu, C. and Kim, Y.-J. (2002) Penalized likelihood regression: general formulation and efficient approximation. *Can. J. Statist.*, **30**, 619–628.

Hall, P. and Opsomer, J. D. (2005) Theory for penalised spline regression. *Biometrika*, **92**, 105–118.

Härdle, W., Hall, P. and Marron, J. S. (1988) How far are automatically chosen regression smoothing parameters from their optimum? *J. Am. Statist. Ass.*, **83**, 86–95.

Harville, D. A. (1977) Maximum likelihood approaches to variance component estimation and to related problems. *J. Am. Statist. Ass.*, **72**, 320–338.

Harville, D. A. (1997) *Matrix Algebra from a Statistician's Perspective*. New York: Springer.

Hastie, T. and Tibshirani, R. (1986) Generalized additive models (with discussion). *Statist. Sci.*, **1**, 297–318.

Hastie, T. and Tibshirani, R. (1990) *Generalized Additive Models*. London: Chapman and Hall.

Hastie, T. and Tibshirani, R. (1993) Varying-coefficient models (with discussion). *J. R. Statist. Soc.* B, **55**, 757–796.

Hurvich, C. M., Simonoff, J. S. and Tsai, C.-L. (1998) Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *J. R. Statist. Soc.* B, **60**, 271–293.

Kalivas, J. H. (1997) Two data sets of near infrared spectra. *Chemometr. Intell. Lab. Syst.*, **37**, 255–259.

Kauermann, G. (2005) A note on smoothing parameter selection for penalized spline smoothing. *J. Statist. Planng Inf.*, **127**, 53–69.

Kauermann, G., Krivobokova, T. and Fahrmeir, L. (2009) Some asymptotic results on generalized penalized spline smoothing. *J. R. Statist. Soc.* B, **71**, 487–503.

Kimeldorf, G. and Wahba, G. (1970) A correspondence between Bayesian estimation of stochastic processes and smoothing by splines. *Ann. Math. Statist.*, **41**, 495–502.

Kohn, R., Ansley, C. F. and Tharm, D. (1991) The performance of cross-validation and maximum likelihood estimators of spline smoothing parameters. *J. Am. Statist. Ass.*, **86**, 1042–1050.

Krivobokova, T., Crainiceanu, C. M. and Kauermann, G. (2008) Fast adaptive penalized splines. *J. Computnl Graph. Statist.*, **17**, 1–20.

Laird, N. M. and Ware, J. H. (1982) Random-effects models for longitudinal data. *Biometrics*, **38**, 963–974.
Lang, S. and Brezger, A. (2004) Bayesian P-splines. *J. Computnl Graph. Statist.*, **13**, 183–212.
Marx, B. D. and Eilers, P. H. (1998) Direct generalized additive modeling with penalized likelihood. *Computnl Statist. Data Anal.*, **28**, 193–209.
Marx, B. D. and Eilers, P. H. (1999) Generalized linear regression on sampled signals and curves: a P-spline approach. *Technometrics*, **41**, 1–13.
Monahan, J. F. (2001) *Numerical Methods of Statistics*. Cambridge: Cambridge University Press.
Nelder, J. A. and Wedderburn, R. W. M. (1972) Generalized linear models. *J. R. Statist. Soc.* A, **135**, 370–384
Nocedal, J. and Wright, S. J. (2006) *Numerical Optimization*, 2nd edn. New York: Springer.
Parker, R. L. and Rice, J. A. (1985) Discussion on 'Some aspects of the spline smoothing approach to non-parametric regression curve fitting' (by B. W. Silverman). *J. R. Statist. Soc.* B, **47**, 40–42.
Patterson, H. D. and Thompson, R. (1971) Recovery of interblock information when block sizes are unequal. *Biometrika*, **58**, 545–554.
Ramsay, J. O. and Silverman, B. W. (2005) *Functional Data Analysis*. New York: Springer.
R Development Core Team (2008) *R 2.8.1: a Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing.
Reiss, P. T. and Ogden, R. T. (2007) Functional principal component regression and functional partial least squares. *J. Am. Statist. Ass.*, **102**, 984–996.
Reiss, P. T. and Ogden, R. T. (2009) Smoothing parameter selection for a class of semiparametric linear models. *J. R. Statist. Soc.* B, **71**, 505–523.
Rue, H., Martino, S. and Chopin, N. (2009) Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations (with discussion). *J. R. Statist. Soc.* B, **71**, 319–392.
Ruppert, D., Wand, M. P. and Carroll, R. J. (2003) *Semiparametric Regression*. Cambridge: Cambridge University Press.
Silverman, B. W. (1985) Some aspects of the spline smoothing approach to non-parametric regression curve fitting (with discussion). *J. R. Statist. Soc.* B, **47**, 1–52.
Tutz, G. and Binder, H. (2006) Generalized additive modeling with implicit variable selection by likelihood-based boosting. *Biometrics*, **62**, 961–971.
Tweedie, M. C. K. (1984) An index which distinguishes between some important exponential families. In *Statistics: Applications and New Directions: Proc. Indian Statistical Institute Golden Jubilee Int. Conf*. (eds J. K. Ghosh and J. Roy), pp. 579–604. Calcutta: Indian Statistical Institute.
Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S*, 4th edn. New York: Springer.
Wahba, G. (1980) Spline bases, regularization and generalized cross validation for solving approximation problems with large quantities of noisy data. In *Approximation Theory III* (ed. E. Cheney). London: Academic Press.
Wahba, G. (1983) Bayesian "confidence intervals" for the cross-validated smoothing spline. *J. R. Statist. Soc.* B, **45**, 133–150.
Wahba, G. (1985) A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. *Ann. Statist.*, **13**, 1378–1402.
Wahba, G. (1990) *Spline Models for Observational Data*. Philadelphia: Society for Industrial and Applied Mathematics.
Wahba, G. and Wold, S. (1975) A completely automatic French curve: fitting spline functions by cross-validation. *Communs Statist. Theor. Meth.*, **4**, 125–141.
Watkins, D. S. (1991) *Fundamentals of Matrix Computations*. New York: Wiley.
Wehrens, R. and Mevik, B.-H. (2007) pls: Partial Least Squares Regression (PLSR) and Principal Component Regression (PCR). *R Package Version 2.1-0*. (Available from `http://mevik.net/work/software/pls.html`.)
Wood, S. N. (2003) Thin plate regression splines. *J. R. Statist. Soc.* B, **65**, 95–114.
Wood, S. N. (2004) Stable and efficient multiple smoothing parameter estimation for generalized additive models. *J. Am. Statist. Ass.*, **99**, 673–686.
Wood, S. N. (2006) *Generalized Additive Models: an Introduction with R*. Boca Raton: CRC–Chapman and Hall.
Wood, S. N. (2008) Fast stable direct fitting and smoothness selection for generalized additive models. *J. R. Statist. Soc.* B, **70**, 495–518.